O Computador como instrumento de ensino na música

Trabalho realizado no âmbito da disciplina de
Informática e Educação

Carlos Gomes n.º 6
Hélia Grilo n.º 15

CESE em Direcção Pedagógica e Administração Escolar

O Computador como instrumento de ensino na música

Resumo

O presente artigo pretende apresentar uma primeira abordagem do uso do computador no ensino da música, como instrumento educacional e como instrumento musical. Desta forma, um dos usos propostos do computador no ensino especializado de música, centra-se na sua utilização como uma ferramenta de trabalho. Isto é especialmente relevante em disciplinas como Análise e Técnicas de Composição e Acústica Musical, em que o computador pode ser utilizado como um instrumento de criação musical, nomeadamente em formas de música contemporânea e electroacústica, e como ferramenta de análise e síntese de sons.

Neste sentido de abordagem, começamos por apresentar historicamente o surgimento da música electroacústica no século XX até à inclusão do computador como ferramenta de trabalho criativo nesta, referindo-nos ainda às técnicas usadas para a representação e síntese digital do som. Neste âmbito, é incluído em anexo a listagem de um programa em *QBASIC 1.1* correspondente à implementação de uma linguagem simples de síntese por computador.

Veremos por fim, e num sentido mais tradicionalista, os usos possíveis para o computador em contextos de ensino/aprendizagem musical na disciplina de Educação Musical do 2º e 3º ciclos do ensino básico e ainda nas diversas disciplinas do ensino vocacional de música.

1. Introdução

Desde o início deste século que o desenvolvimento da electrónica esteve associada à tentativa de produzir novos instrumentos musicais. Em meados dos anos 20, a invenção da amplificação eléctrica permitiu a proliferação de novos instrumentos normalmente concebidos sobre o modelo do órgão: o *Aethérophone* de *Léon Thérémine,* o *Sphärophon* de *Jorg Mager,* o *Trautonium* de *Friedrich Trautwein*, as *Ondas Martenot*[1] de *Maurice Martenot*, entre outros. Este último obteve grande difusão em França derivado ao uso feito deste pelo compositor *Olivier Messiaen*. Trata-se de um instrumento monódico, i.e., só consegue tocar um som de cada vez, permitindo efeitos expressivos como flutuações de intensidade, e passagens em *glissando*[2] de uma nota a outra. Outros instrumentos deste tipo foram usados na música de cinema com o intuito de produzir efeitos sonoros especiais.

A procura do instrumento sintético perfeito, imitação do grande órgão[3] ou de um instrumento novo, desviou evidentemente os inventores e músicos do século XX da concepção de novas construções acústicas e mecânicas. Um fenómeno tão determinante como a criação do saxofone em 1846, por *Adolphe Sax*, não tem equivalente no nosso século. Algumas pesquisas instrumentais acústicas ou neo-acústicas, como as dos irmão *Baschet*, nos anos 50, persistiram e tornaram-se mesmo a especialidade de músicos recentes. Mecânicos no seu princípio, estes aparelhos podem sempre aumentar a sua amplitude acústica pelo emprego da amplificação eléctrica e fixar as suas performances pela gravação. Em muitos casos, contudo, trata-se menos de instrumentos que de objectos construídos por si mesmos, como esculturas sonoras.

Nos instrumentos de teclado, como é o caso do cravo, do órgão ou do piano, existe uma relação directa entre o gesto de carregar numa tecla e a emissão de um som pelo instrumento. No entanto, e ao contrário do que sucede com o piano, em que a intensi-

dade do som produzido é relação directa da velocidade/força de ataque, tanto no cravo como no órgão, este é sempre da mesma intensidade independentemente da velocidade/força de ataque exercida no teclado. Actualmente nos instrumentos electrónicos com recurso ao teclado como elemento de comando e interface Homem/máquina, através da medição do tempo que a tecla leva a passar da posição de repouso para a posição activa, é possível influenciar a intensidade do som produzido numa relação directa e dependente da velocidade/força de ataque exercidas.

Mas porque razão se impôs o modelo do Órgão como arquétipo absoluto, explícito ou implícito das investigações modernas, não só no campo dos tradicionais sintetizadores, como também na concepção e implementação de linguagens de síntese computacional, como é o caso da *CSOUND* desenvolvida no *Massachusetts Institute of Technology* por *Barry L. Vercoe*? Talvez porque já no seu tempo este materializava o sonho do instrumento total e multitímbrico. Aliás, no século XIX o Órgão foi concebido como a síntese da própria orquestra e um eventual seu substituto. É o século da construção de grandes Órgãos sinfónicos com vastos recursos tímbricos.

Ilustração 1 - Um estúdio electroacústico moderno onde o computador tem um papel central.

A invenção do computador digital moderno, aliado à rapidez com que se tem efectuado a evolução tecnológica, faz com que este seja hoje em dia um instrumento usado em todos os domínios da actividade humana. Podemos até falar de uma revolução informática no campo da música, pois serve nomeadamente para editar partituras[4], auxiliar na composição ou inclusive elaborar composições estilísticas[5], sintetizar sons, e como suporte de gravação MIDI[6].

Em todo o caso, se o computador melhorou ou enriqueceu a síntese dos sons, em nada esta alterou a concepção geral dos sistemas. A grande revolução produziu-se antes, nos anos 60 e 70, quando o sintetizador pôs à disposição do interprete ou do compositor sons com uma duração e características potencialmente infinitas, cujo percurso eles podem inflectir, que podem ser esculpidos simultaneamente por evoluções contínuas, etc., tornando possíveis outras formas de música em directo, ao invés do diferido como as técnicas de estúdio já possibilitavam desde os anos 50.

Ilustração 2 - Milton Babbitt junto ao Sintetizador da RCA.

Durante os anos 40 e início dos anos 50 o advento do gravador de fita magnética tinha permitido finalmente a composição musical recorrendo a meios electrónicos. Os instrumentos musicais eléctricos como as *Ondas Martenot* e o *Trautonium* haviam permitido apenas a produção de alguns novos timbres, embora *Messiaen* tivesse demonstrado em *Turangalila* como podiam ser singularmente eficazes os sons algo sobrenaturais das *Ondas Martenot*. Também a manipulação sonora através de discos redundara num

impasse, pois as possibilidades eram limitadas, e as técnicas excessivamente complicadas. As fitas, em compensação, davam ao compositor versatilidade e flexibilidade na gravação e montagem dos sons, permitindo-lhe manipular a altura e o ritmo destes através da alteração da velocidade de reprodução, ou ainda sobrepô-los uns aos outros e organizá-los na ordem que deseja-se. Foram estas técnicas que permitiram o verdadeiro nascimento da música electrónica

Edgard Varése (1883-1965), que nas últimas três décadas vinha lutando pela fabricação de meios electrónicos, recebeu o gravador de fita no final da década de 40, e imediatamente começou a recolher material para o novo "som organizado" com que sonhava. As suas experiências com gravações em discos já haviam deixado antever todo um universo sonoro a ser descoberto: à falta de recursos materiais criativos, ele empreendera a exploração desse universo em obras instrumentais, imitando nas *Intégrales* para orquestra (1924-25) o efeito da reprodução em sentido inverso de uma gravação em disco. Agora, no seu primeiro estudo composto com recurso a meios electrónicos, *Déserts* (1949-54), decidiu usar simultaneamente sons gravados e sons instrumentais, alternando fita e orquestra, de tal maneira que o insólito deriva do familiar e ao primeiro retorna.

Ilustração 3 - Robert Moog junto a três dos seus sintetizadores.

Proliferaram rapidamente os estúdios de música electrónica, sobretudo em estações de rádio, onde já havia a disponibilidade de equipamentos. Entre as primeiras a afirmar a autoridade no campo estiveram a *Radiodiffusion Française* em Paris e a *Nordwestdeutscher Rundfunk* em Colónia, cujos estúdios logo se tornariam centros de facções opostas de compositores electrónicos. Em Paris impôs-se a *musique concrète*, composta de sons naturais modificados e reorganizados, em geral derivados dos metais e da água. O director do estúdio da *Radiodiffusion*, *Pierre Schaeffer* (1910), compôs uma série de obras de *musique concrète* várias delas com colaboração de *Pierre Henry* (1927), que o sucederia no cargo. O primeiro trabalho importante dos dois, criado com técnicas de disco, foi a *Synphonie pour un Homme Seul* (1949-50), uma das primeiras obras electrónicas a serem apresentadas em execução pública.

Em Colónia a situação era muito diferente. *Karleinz Stockhausen* (1928) dedicava tanto tempo ao estúdio quanto à composição instrumental, não se preocupando em transformar sons naturais, mas em criar música electrónica "pura" exclusivamente com os novos recursos. O objectivo do grupo de Colónia era o de sintetizar todo e qualquer som a partir de frequências puras, e foi o que *Stockhausen* tentou em *Studie I* (1953), mas a experiência fracassou. As sonoridades puras não se cristalizavam, como ele esperava, e a ideia teve de ser abandonada até ao advento de métodos e equipamento mais sofisticado. Em certo sentido, no entanto, o projecto foi um sucesso, pois *Studie I*, como as outras peças electrónicas iniciais de *Stockhausen*, parece o mais próximo possível da perfeição se considerarmos as técnicas então disponíveis.

A mediação entre as escolas de Paris e Colónia estabeleceu-se em *Gesung der Jünglinge* (1955-56), em que *Stockhausen* utilizou tanto sons naturais como materiais gerados electronicamente. O objectivo pode aqui ter sido de ordem técnica, fundindo extremos até então opostos.

No entanto para *Milton Babbitt* (1916), o valor da música electrónica não estava nos novos sons, mas no superior controle rítmico que se tornava agora possível. Nos

Ensembles for Synthesizer (1962-64), ele valeu-se dessa possibilidade para impulsionar a construção do serialismo rítmico, tendo explorado a dramaticidade dos sons electrónicos em *Philomel* (1963-64). O sintetizador usado por *Babbitt* na concepção destas duas obras era um instrumento único fabricado pela RCA e doado ao Centro de Música Electrónica Columbia/Princeton em Nova Iorque, nada tendo a ver com os aparelhos do tipo associado ao nome de *Robert Moog*. A chegada do sintetizador *Moog* e outros instrumentos semelhantes ao mercado, em 1964, representou uma revolução nas técnicas da música electrónica. Estes novos sintetizadores ofereciam uma enorme variedade de sons, prontos a serem manipulados, bastando para tal regular os controles do instrumento e tocar num teclado ou dispositivo semelhante.

A partir da década de 60, derivado ao aperfeiçoamento constante dos computadores resultando numa melhoria significativa da sua performance, começa-se a desenvolver o seu uso na composição, especialmente nos Estados Unidos, onde muitos compositores, ensinando em universidades, tinham facilmente acesso a este tipo de equipamentos. No entanto, os primeiros trabalhos de síntese sonora por computador foram feitos nos *Bell Telephone Laboratories* por *Max Matthews*. O seu programa *MUSIC IV* tem sido adaptado e desenvolvido por vários compositores de modo a gerar e modificar sons com o auxílio do computador, tendo dado origem aos seus sucessores *MUSIC IV-B*, implementado em *FORTRAN*, *MUSIC V*, *CMUSIC* e *CSOUND*, estes dois últimos implementados em *C*. A filosofia base de concepção da maioria destes programas de síntese continua, no entanto, a ser a mesma da dos sintetizadores analógicos das décadas de 60 e 70, com a definição em secções separadas, estilo *COBOL*, dos instrumentos e da partitura.

2. A linguagem MIDI

A linguagem MIDI pode ser considerada como uma linguagem matemática onde o elemento básico de comunicação é um agregado de números. Cada número será representado por um BYTE[7] podendo assim representar os número inteiros entre 0 e 255[8].

Existem duas grandes categorias de mensagens, as de CHANNEL e as de SYSTEM. As mensagens CHANNEL podem ainda ser subdivididas em VOICE e MODE. As mensagens do tipo VOICE contêm informação do tipo "Começa a tocar uma nota no canal ...", "Para de tocar uma nota no canal ...", etc. As mensagens do tipo MODE contêm informações sobre o canal MIDI e modos de operação do sintetizador.

As mensagens do tipo SYSTEM podem-se subdividir em três subcategorias: COMMON, REAL-TIME, e EXCLUSIVE. As mensagens do tipo COMMON consistem em mensagens de tipo standard entendidas por qualquer sintetizador. As mensagens do tipo REAL-TIME destinam-se à sincronização temporal de vários instrumentos ligados entre si, sendo as mensagens do tipo EXCLUSIVE proprietárias de um determinado sintetizador, e portanto destinadas ao controle de funções específicas variáveis de sintetizador para sintetizador.

No fluxo MIDI de dados, cada BYTE pode ainda ser um STATUS BYTE[9] ou um DATA BYTE[10], consistindo o primeiro num comando e o segundo no seu argumento.

Os instrumentos MIDI são ligados entre si por cabos com fichas DIN de 5 pernos, existindo três tipos de ficha: MIDI OUT, saída, MIDI IN, entrada, e MIDI THRU, passagem. Normalmente liga-se a saída do computador à entrada do primeiro sintetizador e a saída deste à entrada do computador, ligando-se eventualmente a passagem do primeiro sintetizador à entrada de um segundo sintetizador ou caixa de ritmo.

3. A representação digital do som

Uma onda sonora é representada digitalmente por uma sequência de números que representam as “fotografias” instantâneas da sua amplitude. Tomemos por exemplo a seguinte sinusóide representativa de um som puro:
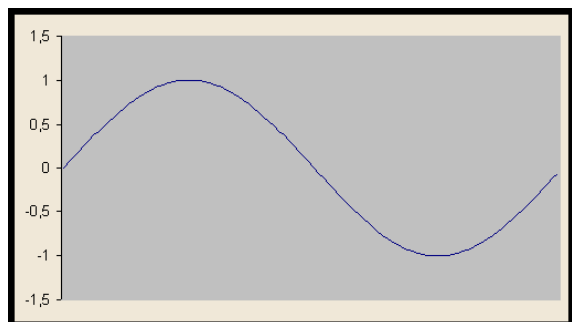
Ilustração 4 - Oscilograma de um som puro.

se pensarmos que esta é representada por uma sequência de pontos, poderemos definir os valores instantâneos de cada um deles numa longa sequência numérica. Vamos supor que tendo esta onda a frequência de 1 Hz, ou seja, um ciclo completo por segundo, retiramo-lhe cem “fotografias” instantâneas da sua amplitude, criando uma sequência numérica a partir destas. Se esta for gravada e posteriormente a lermos rapidamente, consegue-se reproduzir a onda anteriormente registada.

No entanto este sistema, apesar da sua potencial alta fidelidade, possui algumas limitações. Vamos supor que no exemplo acima descrito eram retiradas apenas duas fotografias, uma a meio do ciclo positivo, e outra a meio do ciclo negativo. Repare-se que a onda sonora agora descrita torna-se ia na seguinte:
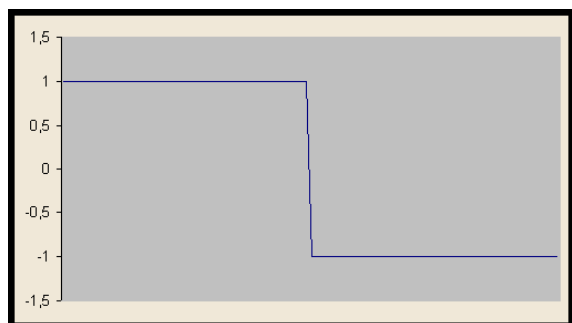
Ilustração 5 - Oscilograma de um som composto por parciais harmónicos ímpares.

ou seja, de uma sinusóide pura sem nenhum harmónico, passaríamos a ter um som rico em harmónicos impares, portanto portador de distorção harmónica, percebida ao nível do timbre do som gravado. Aliás, se for tirada uma única “fotografia” por ciclo, o sinal sonoro passa a ser representado por uma linha contínua, ou seja, de um som gravado de 1 Hz, a resultante passa a ser um som de 0 Hz, portanto, diferente não só timbricamente como também em termos da sua própria frequência[11]. Além deste fenómeno, existem ainda outros efeitos colaterais, como é o facto de se ter na reprodução a frequência de amostragem sobreposta ao sinal gravado, assim como ainda, existir sobreposto a este, a soma e a diferença entre a frequência do som gravado e a frequência de amostragem. Estas frequências parasitas são no entanto facilmente eliminadas por meio do recurso a um filtro passa baixos[12].

4. Teoria básica da síntese sonora

Existem basicamente três processos genéricos de síntese: síntese aditiva, síntese subtractiva, e síntese por distorção. Partindo de um som simples, que é descrito em termos matemáticos pela fórmula:

$$C = SIN(2 \times \Pi \times t \times f)$$

onde f é a frequência deste e t representa o percurso linear do tempo, chegamos facilmente à síntese aditiva que mais não é do que a aplicação do teorema de Fourier[13], donde deduzimos a seguinte fórmula matemática descritiva:

$$C = \sum_{n=1}^{\infty} a_n [SIN(2 \times \Pi \times t \times n \times f)]$$

onde C é o som resultante, a_n é a matriz unidimensional das amplitudes de cada sinusóide, e f é a frequência do som resultante.

A síntese subtractiva, também descritivel em termos matemáticos, consiste em proceder-se de uma forma inversa à da síntese aditiva. Partindo de um som complexo, normalmente ruído branco ou rosa[14], por

meio de processos sucessivos de filtragem, eliminam-se as gamas de frequência indesejáveis. O seguinte modelo matemático transcreve um filtro de primeira ordem, passa baixo ou passa alto[15]:

$$a = \sqrt{[2 \cdot COS(2 \times \Pi \times \pm fc \div Sr)]^2 \cdot 1} \cdot [2 \cdot COS(2 \times \Pi \times \pm fc \div Sr)]$$

$$c_n = (1 \pm a) \times i_n \pm a \times c_{n-1}$$

onde fc é a frequência de corte do filtro, Sr é a frequência de amostragem, i_n é o sinal antes de filtrado, e C_n é o sinal à saída do filtro.

A síntese por distorção faz o recurso a processos de modulação em amplitude e de modulação em frequência. A modulação em amplitude produz duas bandas laterais, onde as frequências c e m produzem ainda as frequências $c+m$ e $c-m$, i.e., de duas frequências produzem-se o total de quatro. Na modulação por frequência, o número de bandas laterais torna-se literalmente infinito, ou seja, além das frequências c e m, temos ainda $c+m$, $c+2m$, $c+3m$, ..., $c+nm$, e ainda as frequências $c-m$, $c-2m$, $c-3m$, ..., $c-nm$. De uma forma matemática poderemos definir a modulação em amplitude por:

$$C = a_m \times [SIN(2 \times \Pi \times t \times fm)] \times a_c \times [SIN(2 \times \Pi \times t \times fc)]$$

e a modulação em frequência por:

$$C = a_c \times [SIN(2 \times \Pi \times t \times a_m \times SIN(2 \times \Pi \times t \times fm) \times fc)]$$

onde C é o som resultante, a_m é a amplitude da moduladora, fm é a frequência da moduladora, a_c é a amplitude da portadora, e fc é a frequência da portadora.

5. O computador como instrumento de ensino

O computador pode ser usado das mais diversas formas em situações de ensino e aprendizagem, podendo constituir uma ferramenta de trabalho de grande valor em diversas áreas disciplinares. Uma das suas utilizações mais obvias parece ser o ensino da informática, nomeadamente a nível da construção de programas, ajudando a estruturar logicamente o raciocínio dos alunos. Assim, a programação poderá ser vista como o meio de resolver problemas significativos que se coloquem aos estudantes servindo simultaneamente como estímulo ao desenvolvimento da sua capacidade lógica e matemática.

Este tipo de actividade, se for bem conduzida, pode ter um elevado valor formativo, nomeadamente porque conduz ao raciocínio algorítmico. A construção dos programas exige dos alunos um esforço complementar para a compreensão dos conceitos, obrigando muitas vezes a encará-los sob novas formas. Problemas possíveis de resultar na implementação de pequenos programas pelos alunos, são, por exemplo, o determinar o máximo divisor comum ou o mínimo múltiplo comum de dois números, averiguar quais os números primos existentes em determinado intervalo numérico, o ordenar uma sucessão de valores por ordem ascendente ou descendente, o calcular a raiz quadrada de um número, etc. Estas são questões vindas da Matemática, podendo o computador ser aqui usado com êxito no ensino de processos algorítmicos, ajudando a desenvolver capacidades de raciocínio lógico.

Ilustração 6 - O *Commodore Amiga* foi um dos primeiros computadores pessoais com possibilidades de multimédia de base.

Para construir programas que resolvam alguns problemas simples referentes a diversas disciplinas, o aluno não precisa conhecer profundamente nenhuma linguagem de programação, nem as mais avançadas, actualmente usadas a nível da programação de aplicações, como o *C++* ou a *JAVA*. Lin-

guagens como o *BASIC*, o *LOGO*, o *PASCAL*, o *PROLOG*, e até o *FORTH*, podem ser usadas com sucesso para a solução de um grande número de problemas, bastando para isso conhecer um número pouco significativo de instruções fundamentais. No entanto, como o objectivo aqui não é fazer do aluno um programador profissional, não deverão talvez existir nestas actividades grandes preocupações com a rapidez ou a elegância dos programas.

No entanto, o ensino da programação tem frequentemente sido encarado numa perspectiva inteiramente diferente, como se tratasse de aprender simplesmente as regras gramaticais e de sintaxe das diversas linguagens de programação. Deve-se ver que a grande utilidade de ensinar a programar um computador, a um nível elementar, não é o de dominar com excelência cada uma das linguagens existentes, mas antes desenvolver as capacidades lógicas e de raciocínio necessárias à resolução de um dado problema. Note-se que de facto a maior dificuldade sentida a nível dos alunos não é o aprender uma gramática ou a sintaxe de uma dada linguagem de programação, mas é a de conseguir elaborar em passos lógicos simples um algoritmo para a solução de um dado problema. Por exemplo, se forem dados a uma pessoa uma sequência de dez números e lhe for pedido que os ordene por ordem ascendente, esta não terá qualquer dificuldade em resolver esta tarefa. No entanto, se lhe for pedido que escreva os passos lógicos que lhe permitiram fazer essa ordenação de forma a que possam ser traduzidos num programa, esta terá de certeza alguma dificuldade em o conseguir fazer, nomeadamente se esta for a primeira vez que o tenta realizar.

O verdadeiro conhecimento não se apoia na memorização de um conjunto infinito de informação. As capacidades limitadas de processamento e tratamento de informação do nosso cérebro isso não permitem. Assim, conhecer é antes de mais saber onde procurar uma dada informação que num dado momento nos é pertinente. Podemos tomar como exemplo a orgânica de uma escola. Estas têm de possuir uma quantidade de dados consideráveis de cada um dos seus alunos, incluindo o seu nome, a sua morada, a sua filiação, relatórios médicos, classificações obtidas em cada uma das disciplinas, etc. Toda esta informação tem de ser periodicamente actualizada sob pena de perder a sua pertinência. Numa escola onde tal sistema informático esteja implementado, é possível poupar tempo do ponto de vista administrativo e burocrático, desde que toda a informação relevante seja introduzida no sistema no momento da sua produção.

Este tipo de trabalho é feito com o recurso a bases de dados. Os alunos podem também organizar as suas próprias bases de dados onde coloquem a informação para eles pertinentes, por exemplo sob a forma de fichas de leitura, permitindo a quando da realização de trabalhos, estes irem buscar directamente a informação pertinente, permitindo assim um uso desta mais eficaz.

Outra das possibilidades no uso do computador são os jogos educacionais. Pode-se aprender muito através de um jogo. Para se ter êxito é necessário estar de alguma maneira activo, identificar e compreender o contexto, assumir um certo grau de responsabilidade pelas consequências, conceber e testar estratégias e modificá-las de acordo com as necessidades. Ao jogador é solicitado aquele tipo de empenhamento pessoal que constitui um dos factores fundamentais em qualquer forma de aprendizagem. No entanto, e apesar de valorizado por alguns educadores, os jogos nunca tiveram um papel significativo no ensino tradicional, nem mesmo nos níveis mais elementares.

O tipo de jogos que prende mais a atenção são os jogos de vídeo dadas as características visuais e dinâmicas da televisão, sendo além disso interactivos. Este último aspecto é particularmente importante na medida em que estudos feitos apontam que é esta questão, aliada ao controlo de uma realidade, o que os torna tão apelativos para as crianças e também para os adultos.

Para que um jogo ou qualquer outra actividade seja considerada educacional, deve ter efeitos relevantes ao nível da aprendizagem e da motivação. Assim, devemos antes do mais distinguir duas categorias de jogos, os concebidos para utilização educacional e os feitos para o grande público, mas que tangencialmente podem encerrar algum potencial educativo. Há jogos para os mais diversos fins educacionais, tais como jogos para desenvolver a leitura, o planeamento, a memória, o raciocínio lógico, a compreensão de mapas e diagramas, etc. Outros jogos podem desenvolver a criatividade, o pensamento divergente, facilitar a aquisição de certos conhecimentos e capacidades.

Muitos dos jogos, especialmente entre os feitos para o grande público, surgem em contextos de violência. Essa violência pode assumir desde formas moderadas, como é a destruição de objectos inanimados, como formas extremas e por vezes até gratuitas, como é o caso da destruição de objectos que representam seres vivos ou até o próprio Homem. No entanto a violência não é um ingrediente indispensável para se ter um bom jogo, sendo esta por vezes importada de outras culturas, como a Japonesa, em que o que para nós se apresenta de uma forma violenta, não mais é para eles do que um ingrediente cultural.

Tradicionalmente, costuma-se dividir o *software* educacional em programas tutoriais e em programas de prática. Os programas tutoriais expõem uma lição segundo uma sequência mais ou menos preestabelecida deixando, em princípio, o aluno avançar ao seu próprio ritmo. Os programas de prática propõem exercícios através dos quais o aluno tem oportunidade de consolidar os seus conhecimentos. Ambos os tipos de programas podem aparecer de forma combinada: ao mesmo tempo que um determinado conceito vai sendo explicado, vão-se propondo questões que testam a sua compreensão pelo aluno e a este só é permitido prosseguir na medida em que vai demonstrando suficiente domínio da matéria já apresentada.

Os programas tutoriais e de prática foram o núcleo do Ensino Assistido por Computador, tendo o seu suporte teórico no ensino programado, que nos anos sessenta chegou a conhecer considerável divulgação, apesar de desde o início ter sido encarado com certas reservas por parte de numerosos educadores, talvez porque não são muitos os objectivos educacionais que se prestam a ser codificados em tutoriais, podendo os programas de prática apenas incidir num tipo muito restrito de competências.

Com o desenvolvimento da Inteligência Artificial surgiu um renovado sopro para os programas tutoriais e de prática. Pensa-se agora em construir sistemas periciais capazes de analisar os erros dos alunos, tomando acções correctivas. Estes sistemas poderiam fazer uso das respostas de cada aluno para irem traçando um quadro global do que este realmente sabe. Assim, possuiriam, não só um modelo da matéria a ensinar, mas também um modelo de quem aprende.

As dificuldades com esta nova fórmula residem no facto de se continuar a representar uma concepção extremamente fechada do conhecimento. Trata-se, ainda, de transmitir qualquer coisa que já existe sem nada de novo se descobrir. Assume-se, assim, a existência de uma verdade oficial, em relação à qual o aluno será sempre incompleto ou defeituoso. Por isso, aprender com este sistema pode acabar de novo por constituir um ensino centrado na máquina ao invés de ser centrado no aluno.

Por último, a totalidade de possibilidades pedagógicas no uso do computador ainda estão por descobrir. É preciso não esquecer que antes do mais este é uma ferramenta a ser usada das mais diversas formas, podendo constituir um orientador/verificador de conhecimentos adquiridos ou ser um instrumento de trabalho sobre as mais diversas formas, desempenhando neste caso um papel de auxiliar pedagógico, ao lado de qualquer outra ferramenta existente na sala de aula.

6. O Computador e a Educação Musical no 2º e 3º ciclo

No âmbito do disposto no despacho normativo n.º 98-A/92, o Ministério da Educação clarificou os objectivos gerais de ciclo para o ensino básico, estabelecendo um quadro de objectivos curriculares que visam a obtenção de competências transversais de várias áreas disciplinares no términos de cada ciclo. Assim, correndo embora o risco de sermos um pouco exaustivos, passamos a apresentar alguns desses objectivos por nos parecerem uma base de sustentação da perspectiva que pretendemos apresentar, sendo simultaneamente elucidativos do ponto de vista da adequação das actividades que posteriormente mencionaremos:

OBJECTIVOS GERAIS DE CICLO	
COMUNICAÇÃO/EXPRESSÃO	
2ºCICLO	**3º CICLO**
• Utilizar técnicas de comunicação verbal, icónica, musical e corporal (diálogo, debate, carta, narração, relato informativo, fotografia, banda desenhada, etc.), respeitando as regras básicas inerentes.	• Utilizar técnicas de comunicação verbal, icónica, musical e corporal (diálogo, debate, exposição oral, mesa redonda; narração, descrição, relatório, apreciação crítica, cartaz, fotografia, vídeo, etc.), respeitando as regras inerentes.
• Interpretar mensagens simples em diferentes códigos de linguagem simbólica e gráfica (expressões matemáticas, gráficos, tabelas, representações, cartográficas, barras cronológicas, esquemas, esboços, vistas ortogonais, notações musicais, sinaléticas de regulação das actividades físicas, etc.) e utilizar convenções e regras básicas dessas linguagens para representar factos, formas, relações e processos elementares.	• Interpretar mensagens em diferentes códigos de linguagem simbólica e gráfica (expressões matemáticas, fórmulas, tabelas, gráficos, diagramas, representações cartográficas, esquemas, esboços, organigramas, vistas ortogonais, notações musicais, sinaléticas de regulação das actividades físicas, etc.) e representar factos, formas, relações e processos, mediante a utilização adequada desses códigos.
• Comunicar criativamente factos, sentimentos, vivências, utilizando diferentes recursos e aplicando técnicas básicas das linguagens verbal, plástica, musical, corporal e dramática.	• Comunicar criativamente ideias, sentimentos, vivências, seleccionando os meios em função da intenção comunicativa, dominando técnicas de expressão das linguagens verbal, plástica, corporal, dramática ou musical.
• Revelar sensibilidade às qualidades (formais, expressivas e físicas) do envolvimento, dos objectos e dos materiais.	• Exprimir criticamente a sua relação com o envolvimento, revelando sensibilidade às suas qualidades físicas, estéticas e funcionais.
• Manifestar sensibilidade estética no exercício de actividades criativas e perante formas de expressão artística (verbais, plásticas, musicais, dramáticas, corporais), quer da cultura em que está inserido quer de povos e épocas diferentes, exprimindo as suas preferências.	• Evidenciar sensibilidade estética no exercício de actividades criativas e na apreciação de manifestações artísticas (literárias, plásticas, musicais, dramáticas, corporais), quer da cultura em que está inserido quer da de povos e épocas diferentes, utilizando as suas experiências e os conhecimentos adquiridos para fundamentar juízos pessoais.
• Recolher, reproduzir e/ou recriar produções do património tradicional português (literatura oral, artesanato, música, jogos).	• Recolher, reproduzir e/ou recriar produções do património tradicional português (literatura oral, artesanato, música, jogos), assumindo-se como elo na cadeia de transmissão desse património.
• Utilizar diferentes documentos de referência (dicionários, gramáticas, registos audiovisuais, dados informáticos, etc.) servindo-se, no caso de documentos impressos, de índices, títulos, ilustrações, entre outros, para localizar e recolher informação necessária à construção das aprendizagens e à concretização de projectos de trabalho.	• Seleccionar e consultar em centros de recursos, bibliotecas, museus, laboratórios, empresas e outras instituições, documentos de natureza diversa (dicionários, gramáticas, protocolos, estatísticas, instruções, registos audiovisuais, dados informáticos, etc.) para recolher informação necessária à concretização de projectos de trabalho.

OBJECTIVOS GERAIS DE CICLO	
COMUNICAÇÃO/EXPRESSÃO	
2ºCICLO	3º CICLO
• Identificar, e resolver, problemas simples e de natureza concreta, explorando soluções alternativas, escolhendo estratégias de resolução adequadas que envolvam a mobilização da informação e dos recursos disponíveis e a análise crítica dos resultados.	• Identificar, e resolver, problemas simples e de natureza concreta, explorando soluções alternativas, escolhendo estratégias de resolução adequadas que envolvam a mobilização da informação e dos recursos disponíveis e a análise crítica dos resultados. • Planear, e realizar, actividades experimentais ou de pesquisa, utilizando as técnicas e os processos apropriados, mobilizando os conteúdos teóricos necessários à formulação de hipóteses explicativas e ao desenvolvimento de procedimentos de verificação que integrem a previsão e a análise crítica dos resultados. • Conhecer características musicais de diferentes géneros, épocas e culturas, de forma a criar/interpretar pequenas composições respeitando a realização vocal ou instrumental desses mesmos géneros e estilos.
• Identificar características da música portuguesa.	• Caracterizar a música portuguesa numa perspectiva histórica, geográfica e social.
• Conhecer diversos instrumentos, com vista à sua utilização expressiva, de acordo com as características sonoras que apresentam.	• Identificar instrumentos das diferentes épocas e conhecer a sua evolução através dos tempos.
• Utilizar técnicas elementares de produção sonora a nível vocal, instrumental e tecnológico.	• Evidenciar domínio de diferentes técnicas de produção sonora a nível vocal, instrumental e tecnológico.

A perspectiva transdisciplinar, marcadamente voltada para formas de ensino independentizadas e vivenciadas em aspectos de carácter prático, com uma forte motivação à pesquisa, está visivelmente patente nestes objectivos cujo alcance é tido como desejável no final de cada ciclo.

Contextualisando desta forma o processo de aprendizagem, não é difícil perspectivar o recurso ao computador, ele próprio um instrumento de trabalho utilitário, multifacetado e multidisciplinar, como um manancial de variadíssimas potencialidades, tendo também a vantagem da sua modernidade ser um aliciante junto dos jovens.

7. Actividades e recursos Informáticos em Educação Musical

No 2º e 3º ciclos de ensino básico a utilização do computador para a prática de alguns conteúdos programáticos é já prevista através de projectos escolares como o Minerva. Existem inclusivamente alguns manuais escolares com unidades programáticas planificadas com recurso à informática. Por exemplo, na disciplina de Educação Musical, um programa como o *Music Works* para Macintosh[16], permite aos alunos:

- escrever música;
- ouvir a música por eles escrita;
- escolher o *andamento* e a *dinâmica* com que esta é tocada;
- seleccionar partes da música, fazendo ouvir uma a uma as várias vozes de uma polifonia;
- seleccionar fontes sonoras, i.e., vários timbres.

Ainda no âmbito da iniciação à notação musical existem outros tipos de recursos

como sejam as enciclopédias CD-ROM onde se pode obter exemplos ilustrativos e informativos. Apresentamos seguidamente um desses exemplos:

Ilustração 7 - Exemplo retirado da enciclopédia em CD-ROM *Encarta 97*.

No que respeita ao treino auditivo existem também possibilidades muito vantajosas para esta aprendizagem que muitas vezes é de difícil estudo individual. Apresentamos um exemplo retirado da mesma enciclopédia relativamente ao parâmetro de altura:

Ilustração 8 - Exemplo de aplicação para o desenvolvimento da associação entre audição e a escrita musical.

Em conteúdos programáticos relacionados com o estudo e a pesquisa organológica, as enciclopédias CD-ROM continuam a ser óptimas ilustrações, históricas e auditivas, permitindo desta forma levar às escolas, com a maior das facilidades, Orquestras Sinfónicas, Ensembles de Música de Câmara, Bandas, etc. Têm ainda a particularidade de apresentar cada instrumento individualmente dando a possibilidade de proporcionar um treino auditivo que faculta ao aluno a identificação, numa obra de conjunto, de um determinado instrumento que a integra.

Ilustração 9 - Instrumento musical (Clarinete).

Desta forma, vemos que o computador pode ser utilizado como forma de desenvolver os diversos parâmetros da audição: altura,

timbre, duração e intensidade. Neste exemplo, retirado da enciclopédia *Encarta 97*, abordámos só os dois primeiros. No entanto estes são sem dúvida aqueles em que o uso do computador pode-se tornar mais premente uma vez que são os que levantam um maior número de dificuldades de assimilação pelos alunos dada a sua intrínseca dificuldade.

8. O computador e o ensino vocacional de música

Apesar do ainda restrito uso do computador como ferramenta educacional no ensino vocacional de música, vemos que este possui algumas potencialidades que podem ser utilizadas neste tipo de ensino. Não considerando as disciplinas de Instrumento/Canto e Classes de Conjunto, as restantes disciplinas, Formação Musical, História da Música, Acústica Musical e Análise e Técnicas de Composição prestam-se à introdução do uso de meios informáticos.

Na disciplina de Formação Musical, o computador pode ser utilizado como ferramenta individual de estudo auditivo. O computador toca uma sequência de sons os quais o aluno tem de escrever em notação musical. Após isto o computador pode apresentar escrito o que tocou, ou através do uso de sistemas periciais, pode até corrigir o que o aluno escreveu, através de um *Scanner* e de um programa de reconhecimento de escrita musical[17]. O Computador ainda pode ser usado na disciplina de Formação Musical como forma de ensinar a teoria musical, através do uso de programas tutoriais, de prática, ou mesmo periciais que combinem ambas as funções.

Na disciplina de História da Música, o uso do computador resumir-se-á talvez ao estilo mais tradicional, sendo adequado ao uso nesta de programas tutoriais e eventualmente de prática. Existem ainda, neste domínio, exemplos de utilização de meios informáticos na implementação de enciclopédias musicais. A título de exemplo, e apesar da sua extrema limitação temática e de aprofundamento das matérias, não se prestando por isso à sua utilização nesta disciplina, o Diário de Notícias editou aproximadamente há um ano atrás a «Enciclopédia DN de Música Clássica», num conjunto de 20 CD's que incluem elementos bibliográficos e excertos de obras de outros tantos autores.

O principal uso do computador na disciplina de Acústica Musical centra-se em torno do estudo do objecto sonoro em termos de análise (oscilograma, espectrograma) e síntese do som (aditiva, subtractiva e por distorção). Aqui o computador equipado com o *software* necessário e uma placa de som pode ser utilizado como um laboratório integrado de acústica, permitindo a análise de sons, a sínteses destes, e até a produção de certos efeitos acústicos, como é o caso da reverberação e do eco, com o intuito de levar o aluno a melhor compreender os fenómenos acústicos envolvidos.

Por último, na disciplina de Análise e Técnicas de Composição, cujos objectivos gerais são o estudo estilístico do repertório da música ocidental a nível da análise e escrita musical, as potencialidades do uso do computador poderão ser múltiplas. Por exemplo, nos Estados Unidos da América o computador é frequentemente utilizado em termos de análise musical, nomeadamente no estudo estatístico de determinadas características musicais. Apesar de neste campo o computador poder ser uma ferramenta por si limitada, visto em termos analíticos não se poder abordar todo o repertório, de pouco mais de 1000 anos, da mesma óptica, é possível utiliza-lo como ferramenta de trabalho com o intuito de pesquisar a recorrência de determinadas características musicais pré-determinadas de carácter estilístico. Simultaneamente, o computador pode ser utilizado como ferramenta de composição, quer seja na reprodução estilística (por exemplo, escrever um moteto ao estilo de Palestrina ou um coral ao estilo de J. S. Bach), como ser utilizado

com um interesse acrescentado na música contemporânea. Desde o uso deste como sequenciador/gravador MIDI, até à composição directa por computador são aplicações que desde a década de 60 vêm sendo desenvolvidas. A este propósito, veja-se o projecto da Universidade da Coimbra, em conjunção com o Conservatório de Coimbra, que desenvolveram em conjunto um programa de composição estilística, SICOM (Sistema Inteligente de Composição Musical), capaz de, através da aprendizagem de excertos de obras de um compositor, compor ao estilo deste. Encontra-se disponível na *Internet*, nos *site*'s da Universidade de Coimbra e do Conservatório de Coimbra, informação sobre este projecto.

Numa perspectiva mais tradicionalista, na música electroacústica, o computador é parte integrante obrigatória de qualquer estúdio moderno, sendo até por vezes este o centro deste.

Por último, o computador pode ser utilizado como instrumento de composição musical, numa estética mais *Avant Gard*, nomeadamente como ferramenta composicional através do recurso a linguagens computacionais de síntese, como as já clássicas *MUSIC V* ou *CMUSIC*.

Um uso mais simplista do computador, como meio de execução musical de uma partitura, não parece possuir grande interesse nesta modalidade de ensino, porque se por um lado pode permitir os alunos executarem facilmente as suas próprias composições musicais, ou até ser usado como instrumentista acompanhador, o seu uso pode fazer com que o aluno deixe de exercitar as suas capacidades de instrumentista e leitura musical, não substituindo as possibilidades e potencialidades interpretativas de um músico. Pode ser que um dia o desenvolvimento em termos da Inteligência Artificial o permita, no entanto o uso das novas tecnologias não tem como objectivo substituir aqui o músico, mas tão somente ser uma ferramenta auxiliar deste.

ANEXO [A]

O exemplo seguinte é uma adaptação de um artigo constante na enciclopédia CD-ROM *Encarta 97*, em que para além de uma explicação teórica sobre o Clarinete, podemos ouvir três exemplos da utilização deste.

Os Primeiros Clarinetes

- *Concerto para Clarinete e Orquestra KV622 (2º andamento) de Mozart*
- *Três peças para Clarinete Solo (1ª peça) de Stravinsky*
- *Raposodia para Orquestra e Clarinete Principal de Debussy*

***Clarinete**, instrumento musical da família dos sopros de madeira, é essencialmente um tubo de corpo cilíndrico colocado em vibração por uma palheta simples que incorpora a embocadura situada no extremo superior do tubo, o extremo inferior termina em forma de sino. Os clarinetes modernos típicos são feitos de ébano (nalguns casos plástico) e têm 20 ou mais perfurações para produzirem sons de diferentes alturas; alguns estão abertos para serem tocados directamente pelo instrumentista, outros estão cobertos por chaves.*

O mais comum dos clarinetes, o soprano em Si b, tem uma extensão de três oitavas e meia; a nota mais grave que emite é o Ré abaixo do Dó central. Parte da extensão sonora deste instrumento é obtida pelo aumento de pressão no sopro (aumentando a velocidade da coluna de ar no instrumento), que causa um aumento da frequência vibratória da coluna de ar. O clarinete, sendo um tubo cilíndrico fechado produz sons a intervalo de 12ª acima da fundamental (ao contrário das flautas e oboés, que produzem sons à oitava acima da fundamental). Os restantes elementos da família mais comum dos Clarinetes são o Soprano em Lá; O Alto em Mi b; o Baixo (oitava abaixo do soprano); e o Contrabaixo (oitava abaixo do Baixo). O percursor do Clarinete Alto nos finais do século XVIII foi o Cor de Basset. A música para todos os Clarinetes é escrita como se o Clarinete fosse em Dó; num Clarinete em Si b um Dó escrito na pauta soa como um Si b. Os instrumentistas podem assim trocar de instrumento sem terem que aprender novas dedilhações. O termo Si b refere-se à afinação do instrumento e não ao som da sua nota fundamental.

Clarinete foi inventado por volta de 1700 pelo Germânico construtor de flautas Johann Christoph Denner como modificação da popular Chalumeau. Cerca de 1840 foram desenvolvidos dois complexos sistemas de dedilhação: o sistema Boehm, usado na maioria dos países, e patenteado em 1844 pelo construtor francês Auguste Buffet, que adaptou à flauta os melhoramentos do construtor alemão Theobald Boehm; e o sistema de perfuração mais estreita e de som mais velado desenvolvido cerca de 1860 pelo construtor belga Eugène Albert.

Os clarinetes tornaram-se parte integrante das orquestras cerca de 1780. As primeiras obras a visarem este instrumento incluem a Abertura para dois clarinetes e trompa (1748) do compositor de origem Germano-Britânica, George Frideric Haendel e o Concerto para Clarinete (1791) de Wolfgang Amadeus Mozart."

ANEXO [B]

Relativamente à sua filosofia de concepção, SCORE 2.00 é bastante diferente das diversas versões de MUSIC e de CSOUND, sendo estas últimas concebidas de forma a separar em secções de programação distintas a definição dos instrumentos, i.e. sons, e a sua execução. Esta arquitectura assemelha-se ao COBOL, que se divide em quatro secções distintas: *Identification Division*, *Environment Division*, *Data Division*, e *Procedure Division*. Em *SCORE 2.00* a concepção é feita segundo o princípio da montagem directa de sons no meio de gravação, de certa forma à semelhança das técnicas de estúdio usadas por *Pierre Schaffer* no estúdio da *Radiodiffusion*, em País, em que se vão montando sobre o meio de gravação/reprodução objectos sonoros correspondentes à síntese sonora efectuada, procedendo ao seu tratamento através do recurso a filtragens e reverberações.

Sumário dos comandos existentes em SCORE 2.00

< > Indica itens obrigatórios
[ ] Indica Itens opcionais
| Separador entre opções

END

Termina um ficheiro descritor.

FILTER <s_tune> <f_tune> <HIGH|LOW> [CLEAR]

<s_tune> nota inicial de corte do filtro, entre -128 e 127
<f_tune> nota final de corte do filtro, entre -128 e 127
<HIGH|LOW> passa alto (HIGH) ou passa baixo (LOW)
[CLEAR] inicializa o comportamento do filtro

Filtro passa baixo ou passa alto, actuando sobre o *buffer* de secção.

MODULATION <c_0-7> <c_p> <c_st> c_ft> <c_sp> <c_fp> <m_0-7> <m_p> <m_st> <m_ft> <m_sp> <m_fp> <scale>

<c_0-7> forma de onda da portadora definida por WAVE
<c_p> fase da portadora, entre 0 e 99
<c_st> nota inicial da portadora, entre -128 e 127
<c_ft> nota final da portadora, entre -128 e 127
<c_sp> amplitude inicial da portadora, entre 0 e 99
<c_fp> amplitude final da portadora, entre 0 e 99
<m_0-7> forma de onda da moduladora definida por WAVE
<m_p> fase da moduladora, entre 0 e 99
<m_st> nota inicial da moduladora, entre -128 e 127
<m_ft> nota final da moduladora, entre -128 e 127
<m_sp> amplitude inicial da moduladora, entre 0 e 99
<m_fp> amplitude final da moduladora, entre 0 e 99
<scale> valor pelo qual a amplitude final do sinal, dividida antes de ser adicionado ao *buffer* de secção, entre 1 e 10000

Adiciona ao *buffer* de secção uma resultante de FM simples.

NOISE <s_amp> <f_amp> <scale>

<s_amp> amplitude inicial, entre 0 e 99
<f_amp> amplitude final, entre 0 e 99
<scale> valor pelo qual a amplitude final do sinal, dividida antes de ser adicionado ao *buffer* de secção, entre 1 e 10000

Adiciona ao *buffer* de secção ruído branco.

OUTPUT <file> <s_rate> <duration>

<file> nome do ficheiro para saída sem extensão do MSDOS
<s_rate> ratio de amostragem, entre 56 Hz e 65534 Hz
<duration> duração total em segundos

Abre um ficheiro para saída.

RECORD [NOCLEAR]

[NOCLEAR] mantém intacto o *buffer* de secção.

Escreve no ficheiro de saída o *buffer* de secção.

REMARK

Permite escrever um comentário à frente de cada instrução REMARK.

REVERB <s_delay> <f_delay> <s_amp> <f_amp> <scale> [CLEAR]

<s_delay> atraso inicial do sinal em milisegundos
<f_delay> atraso final do sinal em milisegundos
<s_amp> amplitude inicial do sinal reinjectado, entre 0 e 99
<f_amp> amplitude final do sinal reinjectado, entre 0 e 99
<scale> relação com o sinal atrasado e o real; a amplitude do sinal atrasado, de *1/s* e a do sinal real de *(s-1)/s*, com *s* entre 1 e 10000
[CLEAR] inicializa o comportamento da reverberação

Introduz uma reverberação no sinal gravado no *buffer* de secção.

SECTION <length> [SYNC]

<length> tamanho do *buffer* corrente em milisegundos
[SYNC] inicializa os geradores de onda

Define o comprimento do *buffer* de secção.

SOUND <0-7> <phase> <s_tune> <f_tune> <s_amp> <f_amp> <scale>

<0-7> forma de onda definida por WAVE
<phase> fase, entre 0 e 99
<s_tune> nota inicial, entre -128 e 127
<f_tune> nota final, entre -128 e 127
<s_amp> amplitude inicial, entre 0 e 99
<f_amp> amplitude final, entre 0 e 99

<scale> valor pelo qual a amplitude final do sinal, dividida antes de ser adicionado ao *buffer* de secção, entre 1 e 10000

Adiciona ao *buffer* de secção uma onda simples definida por WAVE.

TUNING <base> <step>

<base> frequência base, entre 10 Hz e 10000 Hz
<step> divisão de oitava, entre 1 e 2

Define a frequência base assim como a razão de divisão de oitava.

WAVE <0-7> <CLEAR|SINE|SQUARE <1-99>|RAMP UP|RAMP DOWN|SWALOW>

<0-7> número da forma de onda a ser definida
<CLEAR...> especifica o tipo de onda pretendido

Define até 8 formas de onda diferentes, do tipo acima especificado, que inclui sinusóides, onda quadrada com Duty-Cycle variável, dente de serra e onda triangular.

ANEXO [C]

Um dos usos defendidos, no presente trabalho, do computador no ensino especializado de música, centra-se na utilização deste como uma ferramenta de trabalho. Isto é especialmente relevante em disciplinas como *Análise e Técnicas de Composição* e *Acústica Musical*, em que o computador pode ser utilizado como instrumento de composição, nomeadamente em formas de música contemporânea e electroacústica, e como ferramenta de análise e síntese de sons. Neste sentido, a listagem em *MS-DOS QBasic 1.1*, a seguir transcrita, é a implementação de um programa de síntese musical que pode ser usado como um recurso bastante acessível, visto que corre em qualquer computador compatível correndo MS-DOS, necessitando de uma placa de som de 16 bit's, algo que hoje qualquer computador multimédia possui obrigatoriamente.

```
            REM *****************************************************
            REM *Compilador de ficheiros descritores em «Score 2.00»*
            REM *        (C) 1997 Carlos Alberto F. F. Gomes        *
            REM *****************************************************
            SCREEN 7
            CLEAR , , 8192
            LET dir$ = "C:\Score200"
            LET SxOnly% = 0
            OPEN dir$ + "\buffer.tmp" FOR RANDOM ACCESS READ WRITE AS #4
LEN = 2
            OPEN dir$ + "\reverb.tmp" FOR RANDOM ACCESS READ WRITE AS #5
LEN = 2
            DIM waves%(4095, 7)
            DIM index(99)
            RESTORE ModIndex:
            FOR n% = 0 TO 99
            READ index(n%)
            NEXT n%
            LOCATE 15, 1
            SHELL "dir " + dir$ + "\*.sc /b /l"
            LOCATE 1, 1
            COLOR 6
            PRINT SPACE$(520)
            LOCATE 14, 1
            PRINT "----------------------------------------"
            LOCATE 1, 1
            COLOR 3
            PRINT ">>>>>>>>>>>>>>>Score 2.00<<<<<<<<<<<<<<<"
            PRINT
            PRINT "SoundMaker 16 bit mono version"
            PRINT
Inputfile:  LOCATE 5, 1
            INPUT "Input file? ", file$
            ON ERROR GOTO Error1:
            IF LEFT$(file$, 1) <> "@" THEN GOTO OpenFiles:
            LET SxOnly% = -1
            LET file$ = RIGHT$(file$, LEN(file$) - 1)
OpenFiles:  OPEN dir$ + "\" + file$ + ".sc" FOR INPUT AS #1 LEN = 1024
            OPEN dir$ + "\" + file$ + ".rp" FOR OUTPUT AS #2 LEN = 1024
            IF SxOnly% THEN PRINT #2, "Probe only mode..."
            PRINT #2, "Started on: "; DATE$; " at "; TIME$
            LOCATE 5, 1
            PRINT SPACE$(255)
            LOCATE 10, 1
            PRINT "Started on: "; DATE$; " at "; TIME$
            LET basefreq = 10
```

```
            LET setfreq = 1
            ON ERROR GOTO Error2:
            GOTO MainLoop:

Error1:     LOCATE 5, 1
            PRINT SPACE$(255)
            RESUME Inputfile:

Error2:     PRINT #2, "SYSTEM ERROR at line"; LineNumber&; "["; a$; "]"
            LET errors& = errors& + 1
            RESUME MainLoop:

ModIndex:   DATA 0,0,0,0.001
            DATA 0.001,0.001,0.002,0.002
            DATA 0.002,0.003,0.003,0.004
            DATA 0.005,0.005,0.006,0.008
            DATA 0.008,0.009,0.011,0.012
            DATA 0.014,0.015,0.017,0.018
            DATA 0.020,0.022,0.024,0.026
            DATA 0.028,0.030,0.033,0.036
            DATA 0.040,0.043,0.047,0.051
            DATA 0.056,0.061,0.066,0.072
            DATA 0.079,0.085,0.094,0.103
            DATA 0.112,0.122,0.133,0.145
            DATA 0.158,0.172,0.188,0.205
            DATA 0.224,0.224,0.266,0.290
            DATA 0.316,0.345,0.376,0.410
            DATA 0.447,0.488,0.532,0.580
            DATA 0.632,0.690,0.752,0.820
            DATA 0.894,0.975,1.064,1.160
            DATA 1.265,1.379,1.504,1.640
            DATA 1.789,1.951,2.217,2.320
            DATA 2.530,2.759,3.008,3.281
            DATA 3.578,3.901,4.255,4.640
            DATA 5.060,5.517,6.017,6.561
            DATA 7.155,7.803,8.509,9.279
            DATA 10.119,11.035,12.034,13.123

            REM Interpretador
MainLoop:   LOCATE 5, 1
            PRINT "Done";
            PRINT USING "##########"; TAB(10); CurrentPointer&
            PRINT "out of";
            PRINT USING "##########"; TAB(10); EndPointer&
            LOCATE 8, 9
            LET LineNumber& = LineNumber& + 1
            PRINT "Line:"; LineNumber&; TAB(24); "Errors:"; errors&
            GOSUB NextLine:
            IF UCASE$(b$) = "SOUND" THEN GOTO InsSound:
            IF UCASE$(b$) = "MODULATION" THEN GOTO InsModul:
            IF UCASE$(b$) = "REVERB" THEN GOTO InsReverb:
            IF UCASE$(b$) = "NOISE" THEN GOTO InsNoise:
            IF UCASE$(b$) = "FILTER" THEN GOTO InsFilter:
            IF UCASE$(b$) = "RECORD" THEN GOTO InsRecord:
            IF UCASE$(b$) = "SECTION" THEN GOTO InsSection:
            IF UCASE$(b$) = "WAVE" THEN GOTO InsWave:
            IF UCASE$(b$) = "TUNING" THEN GOTO InsTuning:
            IF UCASE$(b$) = "OUTPUT" THEN GOTO InsOutput:
            IF UCASE$(b$) = "REMARK" OR b$ = "" THEN GOTO MainLoop:
            IF UCASE$(b$) = "END" THEN GOTO InsEnd:
SxError:    PRINT #2, "SINTAX ERROR at line"; LineNumber&; "["; a$; "]"
```

```
            LET errors& = errors& + 1
            GOTO MainLoop:
RevbError:  PRINT #2, "HIGH REVERB TIME at line"; LineNumber&; "["; a$; "]"
            LET errors& = errors& + 1
            GOTO MainLoop:
SrError:    PRINT #2, "LOW SAMPLE RATE at line"; LineNumber&; "["; a$; "]"
            LET errors& = errors& + 1
            GOTO MainLoop:

NextLine:   LET P2% = 1
            IF EOF(1) THEN GOTO InsEnd:
            LINE INPUT #1, a$
NextParm:   LET b$ = ""
            IF LEN(a$) = 0 OR LEN(a$) < P2% THEN RETURN
            FOR P1% = P2% TO LEN(a$)
              IF MID$(a$, P1%, 1) <> " " THEN GOTO ReadStr:
            NEXT P1%
            RETURN
ReadStr:    FOR P2% = P1% TO LEN(a$)
              IF MID$(a$, P2%, 1) = " " THEN RETURN
              LET b$ = b$ + MID$(a$, P2%, 1)
            NEXT P2%
            RETURN

InsRecord:  IF CurrentPointer& + SecLength& > EndPointer& THEN GOTO InsEnd:
            LET CurrentPointer& = CurrentPointer& + SecLength&
            GOSUB NextParm:
            IF UCASE$(b$) = "NOCLEAR" THEN LET flag% = -1 ELSE LET flag% =
0
            LET WavePos& = WavePos& + SecLength&
            IF SxOnly% THEN GOTO MainLoop:
            FOR n& = 1 TO SecLength&
              GET #4, n&, buffer%
              LET HighByte% = buffer% \ 256
              LET LowByte% = buffer% AND 255
              LET buffer% = buffer% AND flag%
              IF HighByte% < 0 THEN LET HighByte% = HighByte% + 256
              PRINT #3, CHR$(LowByte%);
              PRINT #3, CHR$(HighByte%);
              PUT #4, n&, buffer%
            NEXT n&
            GOTO MainLoop:

InsSection: GOSUB NextParm:
            IF b$ = "" THEN GOTO SxError:
            IF VAL(b$) < 1 THEN GOTO SxError:
            LET ReverbSec& = SecLength&
            LET SecLength& = VAL(b$) * SampleRate& / 1000
            IF SxOnly% THEN GOTO MainLoop:
            LET buffer% = 0
            FOR n& = 1 TO SecLength&
              PUT #4, n&, buffer%
            NEXT n&
            IF UCASE$(b$) <> "SYNC" THEN GOTO MainLoop:
            LET WavePos& = 0
            GOTO MainLoop:

InsTuning:  GOSUB NextParm:
            IF b$ = "" THEN GOTO SxError:
            IF VAL(b$) < 10 OR VAL(b$) > 10000 THEN GOTO SxError:
            LET basefreq = VAL(b$)
```

```
            GOSUB NextParm:
            IF b$ = "" THEN GOTO SxError:
            IF VAL(b$) < 1 OR VAL(b$) > 2 THEN GOTO SxError:
            LET stepfreq = VAL(b$)
            GOTO MainLoop:

InsWave:    GOSUB NextParm:
            IF b$ = "" THEN GOTO SxError:
            LET wavenumber% = VAL(b$)
            IF wavenumber% < 0 OR wavenumber% > 7 THEN GOTO SxError:
            GOSUB NextParm:
            IF UCASE$(b$) = "CLEAR" THEN GOTO WaveClear:
            IF UCASE$(b$) = "SINE" THEN GOTO WaveSine:
            IF UCASE$(b$) = "SQUARE" THEN GOTO WaveSquare:
            IF UCASE$(b$) = "RAMP" THEN GOTO WaveRamp:
            IF UCASE$(b$) = "SWALOW" THEN GOTO WaveSwalow:
            GOTO SxError:
WaveClear:  IF SxOnly% THEN GOTO MainLoop:
            FOR n% = 0 TO 4095
              LET waves%(n%, wavenumber%) = 0
            NEXT n%
            GOTO MainLoop:
WaveSine:   IF SxOnly% THEN GOTO MainLoop:
            FOR n% = 0 TO 4095
              LET waves%(n%, wavenumber%) = CINT(SIN(n% * 3.141593 / 2048)
* 32767)
            NEXT n%
            GOTO MainLoop:
WaveSquare: GOSUB NextParm:
            IF b$ = "" THEN GOTO SxError:
            LET DutyCycle% = VAL(b$)
            IF DutyCycle% < 1 OR DutyCycle% > 99 THEN GOTO SxError:
            IF SxOnly% THEN GOTO MainLoop:
            FOR n% = 0 TO 4095
              LET waves%(n%, wavenumber%) = (n% < DutyCycle% * 40.95) *
65535 + 32767
            NEXT n%
            GOTO MainLoop:
WaveRamp:   GOSUB NextParm:
            IF UCASE$(b$) = "UP" THEN GOTO RampUp
            IF UCASE$(b$) = "DOWN" THEN GOTO RampDown:
            GOTO SxError:
RampUp:     IF SxOnly% THEN GOTO MainLoop:
            FOR n% = 0 TO 4095
              LET waves%(n%, wavenumber%) = n% * 16# - 32768
            NEXT n%
            GOTO MainLoop:
RampDown:   IF SxOnly% THEN GOTO MainLoop:
            FOR n% = 0 TO 4095
              LET waves%(n%, wavenumber%) = (4095 - n%) * 16# - 32768
            NEXT n%
            GOTO MainLoop:
WaveSwalow: IF SxOnly% THEN GOTO MainLoop:
            FOR n% = 0 TO 1023
              LET waves%(n%, wavenumber%) = n% * 32
            NEXT n%
            FOR n% = 1024 TO 3071
              LET waves%(n%, wavenumber%) = (1023 - n% + 1024) * 32
            NEXT n%
            FOR n% = 3072 TO 4095
              LET waves%(n%, wavenumber%) = (n% - 4095) * 32
```

```
            NEXT n%
            GOTO MainLoop:

InsOutput:  GOSUB NextParm:
            IF b$ = "" THEN GOTO SxError:
            LET name$ = b$
            GOSUB NextParm:
            IF b$ = "" THEN GOTO SxError:
            LET SampleRate& = VAL(b$)
            IF SampleRate& < 56 OR SapleRate& > 65534 THEN GOTO SxError:
            GOSUB NextParm:
            IF b$ = "" THEN GOTO SxError:
            LET seconds = VAL(b$)
            IF seconds < 0 THEN GOTO SxError:
            LET EndPointer& = SampleRate& * seconds
            PRINT #2, "Sound File: "; name$ + ".wav"
            PRINT #2, "Sample Rate:"; SampleRate&; "Hz"
            PRINT #2, "Duration:"; seconds; "seconds"
            IF SxOnly% THEN GOTO MainLoop:
            OPEN dir$ + "\" + name$ + ".wav" FOR OUTPUT AS #3 LEN = 1024
            PRINT #3, "RIFF";
            LET Num32& = EndPointer& * 2 + 36
            GOSUB PrtNum32:
            PRINT #3, "WAVEfmt ";
            PRINT #3, CHR$(16); STRING$(3, 0);
            PRINT #3, CHR$(1); CHR$(0); CHR$(1); CHR$(0);
            LET Num32& = SampleRate&
            GOSUB PrtNum32:
            LET Num32& = SampleRate& * 2
            GOSUB PrtNum32:
            PRINT #3, CHR$(2); CHR$(0); CHR$(16); CHR$(0);
            PRINT #3, "data";
            LET Num32& = EndPointer& * 2
            GOSUB PrtNum32:
            GOTO MainLoop:
PrtNum32:   PRINT #3, CHR$(Num32& AND 255);
            PRINT #3, CHR$(Num32& \ 256 AND 255);
            PRINT #3, CHR$(Num32& \ 65536 AND 255);
            PRINT #3, CHR$(Num32& \ 16777216 AND 255);
            RETURN

InsSound:   GOSUB NextParm:
            IF b$ = "" THEN GOTO SxError:
            LET wavenumber% = VAL(b$)
            IF wavenumber% < 0 OR wavenumber% > 7 THEN GOTO SxError:
            GOSUB NextParm:
            IF b$ = "" THEN GOTO SxError:
            LET phase% = VAL(b$)
            IF phase% < 0 OR phase% > 99 THEN GOTO SxError:
            GOSUB NextParm:
            IF b$ = "" THEN GOTO SxError:
            LET stune% = VAL(b$)
            IF stune% < -128 OR stune% > 127 THEN GOTO SxError:
            GOSUB NextParm:
            IF b$ = "" THEN GOTO SxError:
            LET ftune% = VAL(b$)
            IF ftune% < -128 OR ftune% > 127 THEN GOTO SxError:
            GOSUB NextParm:
            IF b$ = "" THEN GOTO SxError:
            LET samp% = VAL(b$)
            IF samp% < 0 OR samp% > 99 THEN GOTO SxError:
```

```
            GOSUB NextParm:
            IF b$ = "" THEN GOTO SxError:
            LET famp% = VAL(b$)
            IF famp% < 0 OR famp% > 99 THEN GOTO SxError:
            GOSUB NextParm:
            IF b$ = "" THEN GOTO SxError:
            LET scale% = VAL(b$)
            IF scale% < 1 OR scale% > 10000 THEN GOTO SxError:
            LET ainc% = famp% - samp%
            LET sfreq = basefreq * stepfreq ^ stune%
            LET finc = basefreq * stepfreq ^ ftune% - sfreq
            IF sfreq > SampleRate& / 2 OR sfreq + finc > SampleRate& / 2
THEN GOTO SrError:
            IF SxOnly% THEN GOTO MainLoop:
            FOR n& = 1 TO SecLength&
              LET samples = SampleRate& / (sfreq + finc / SecLength& * n&)
              LET position% = (n& + WavePos&) / samples * 4096 AND 4095
              LET v% = waves%(position% + phase% / 99 * 4095 AND 4095,
wavenumber%)
              LET v% = v% * (samp% + ainc% * n& / SecLength&) / 99
              GET #4, n&, buffer%
              LET buffer% = buffer% + v% \ scale%
              PUT #4, n&, buffer%
            NEXT n&
            GOTO MainLoop:

InsModul:   GOSUB NextParm:
            IF b$ = "" THEN GOTO SxError:
            LET cwavenumber% = VAL(b$)
            IF cwavenumber% < 0 OR cwavenumber% > 7 THEN GOTO SxError:
            GOSUB NextParm:
            IF b$ = "" THEN GOTO SxError:
            LET cphase% = VAL(b$)
            IF cphase% < 0 OR cphase% > 99 THEN GOTO SxError:
            GOSUB NextParm:
            IF b$ = "" THEN GOTO SxError:
            LET cstune% = VAL(b$)
            IF cstune% < -128 OR cstune% > 127 THEN GOTO SxError:
            GOSUB NextParm:
            IF b$ = "" THEN GOTO SxError:
            LET cftune% = VAL(b$)
            IF cftune% < -128 OR cftune% > 127 THEN GOTO SxError:
            GOSUB NextParm:
            IF b$ = "" THEN GOTO SxError:
            LET csamp% = VAL(b$)
            IF csamp% < 0 OR csamp% > 99 THEN GOTO SxError:
            GOSUB NextParm:
            IF b$ = "" THEN GOTO SxError:
            LET cfamp% = VAL(b$)
            IF cfamp% < 0 OR cfamp% > 99 THEN GOTO SxError:
            LET cinc% = cfamp% - csamp%
            LET csfreq = basefreq * stepfreq ^ cstune%
            LET cfinc = basefreq * stepfreq ^ cftune% - csfreq
            IF csfreq > SampleRate& / 2 OR csfreq + cfinc > SampleRate& / 2
THEN GOTO SrError:
            GOSUB NextParm:
            IF b$ = "" THEN GOTO SxError:
            LET mwavenumber% = VAL(b$)
            IF mwavenumber% < 0 OR mwavenumber% > 7 THEN GOTO SxError:
            GOSUB NextParm:
            IF b$ = "" THEN GOTO SxError:
```

```
            LET mphase% = VAL(b$)
            IF mphase% < 0 OR mphase% > 99 THEN GOTO SxError:
            GOSUB NextParm:
            IF b$ = "" THEN GOTO SxError:
            LET mstune% = VAL(b$)
            IF mstune% < -128 OR mstune% > 127 THEN GOTO SxError:
            GOSUB NextParm:
            IF b$ = "" THEN GOTO SxError:
            LET mftune% = VAL(b$)
            IF mftune% < -128 OR mftune% > 127 THEN GOTO SxError:
            GOSUB NextParm:
            IF b$ = "" THEN GOTO SxError:
            LET msamp% = VAL(b$)
            IF msamp% < 0 OR msamp% > 99 THEN GOTO SxError:
            GOSUB NextParm:
            IF b$ = "" THEN GOTO SxError:
            LET mfamp% = VAL(b$)
            IF mfamp% < 0 OR mfamp% > 99 THEN GOTO SxError:
            GOSUB NextParm:
            IF b$ = "" THEN GOTO SxError:
            LET scale% = VAL(b$)
            IF scale% < 1 OR scale% > 10000 THEN GOTO SxError:
            LET minc% = mfamp% - msamp%
            LET msfreq = basefreq * stepfreq ^ mstune%
            LET mfinc = basefreq * stepfreq ^ mftune% - msfreq
            IF msfreq > SampleRate& / 2 OR msfreq + mfinc > SampleRate& / 2
THEN GOTO SrError:
            IF SxOnly% THEN GOTO MainLoop:
            FOR n& = 1 TO SecLength&
              LET msamples = SampleRate& / (msfreq + mfinc / SecLength& *
n&)
              LET mposition% = (n& + WavePos&) / msamples * 4096 AND 4095
              LET m& = waves%(mposition% + mphase% / 99 * 4095 AND 4095,
mwavenumber%)
              LET m& = m& * index(msamp% + minc% / SecLength& * n&)
              LET csamples = SampleRate& / (csfreq + cfinc / SecLength& *
n&)
              LET cposition% = ((n& + WavePos&) / csamples * 4096 AND 4095)
+ (m& \ 8 AND 4095)
              LET v% = waves%(cposition% + phase% / 99 * 4095 AND 4095,
cwavenumber%)
              LET v% = v% * (csamp% + cinc% * n& / SecLength&) / 99
              GET #4, n&, buffer%
              LET buffer% = buffer% + v% \ scale%
              PUT #4, n&, buffer%
            NEXT n&
            GOTO MainLoop:

InsNoise:   GOSUB NextParm:
            IF b$ = "" THEN GOTO SxError:
            LET samp% = VAL(b$)
            IF samp% < 0 OR samp% > 99 THEN GOTO SxError:
            GOSUB NextParm:
            IF b$ = "" THEN GOTO SxError:
            LET famp% = VAL(b$)
            IF famp% < 0 OR famp% > 99 THEN GOTO SxError:
            GOSUB NextParm:
            IF b$ = "" THEN GOTO SxError:
            LET scale% = VAL(b$)
            IF scale% < 1 OR scale% > 10000 THEN GOTO SxError:
            LET ainc% = famp% - samp%
```

```
            IF SxOnly% THEN GOTO MainLoop:
            FOR n& = 1 TO SecLength&
              LET v% = RND * 65535 - 32768
              LET v% = v% * (samp% + ainc% * n& / SecLength&) / 99
              GET #4, n&, buffer%
              LET buffer% = buffer% + v% \ scale%
              PUT #4, n&, buffer%
            NEXT n&
            GOTO MainLoop:

InsFilter:  GOSUB NextParm:
            IF b$ = "" THEN GOTO SxError:
            LET stune% = VAL(b$)
            IF stune% < -128 OR stune% > 127 THEN GOTO SxError:
            GOSUB NextParm:
            IF b$ = "" THEN GOTO SxError:
            LET ftune% = VAL(b$)
            IF ftune% < -128 OR ftune% > 127 THEN GOTO SxError:
            GOSUB NextParm:
            IF UCASE$(b$) = "HIGH" THEN GOTO High:
            IF UCASE$(b$) = "LOW" THEN GOTO Low:
            GOTO SxError:
High:       LET flag% = -1
            GOTO Filter:
Low:        LET flag% = 1
Filter:     GOSUB NextParm:
            IF UCASE$(b$) = "CLEAR" THEN LET sample((flag% + 1) \ 2) = 0
            LET SFc = basefreq * stepfreq ^ stune% * flag%
            LET finc = basefreq * stepfreq ^ ftune% * flag% - SFc
            IF SFc > SampleRate& / 2 OR SFc + finc > SampleRate& / 2 THEN
GOTO SrError:
            IF SxOnly% THEN GOTO MainLoop:
            FOR n& = 1 TO SecLength&
              LET SFr = SFc + finc * n& / SecLength&
              LET v1 = 2 - COS(6.283185 * SFr / SampleRate&)
              LET v2 = SQR(v1 * v1 - 1) - v1
              LET v3 = SFr < 0 OR -(SFr >= 0)
              GET #4, n&, buffer%
              LET sample((flag% + 1) \ 2) = (v2 + 1) * buffer% - v2 * v3 *
sample((flag% + 1) \ 2)
              LET buffer% = sample((flag% + 1) \ 2)
              PUT #4, n&, buffer%
            NEXT n&
            GOTO MainLoop:

InsReverb:  GOSUB NextParm:
            IF b$ = "" THEN GOTO SxError:
            LET cdelay& = VAL(b$)
            IF cdelay& < 0 THEN GOTO SxError:
            GOSUB NextParm:
            IF b$ = "" THEN GOTO SxError:
            LET fdelay& = VAL(b$)
            IF fdelay& < 0 THEN GOTO SxError:
            GOSUB NextParm:
            IF b$ = "" THEN GOTO SxError:
            LET samp% = VAL(b$)
            IF samp% < 0 OR samp% > 99 THEN GOTO SxError:
            GOSUB NextParm:
            IF b$ = "" THEN GOTO SxError:
            LET famp% = VAL(b$)
            IF famp% < 0 OR famp% > 99 THEN GOTO SxError:
```

```
            GOSUB NextParm:
            IF b$ = "" THEN GOTO SxError:
            LET scale% = VAL(b$)
            IF scale% < 1 OR scale% > 10000 THEN GOTO SxError:
            GOSUB NextParm:
            IF UCASE$(b$) <> "CLEAR" THEN GOTO Revrb:
            LET ReverbSec& = SecLength&
            IF SxOnly% THEN GOTO MainLoop:
            LET reverb% = 0
            FOR n& = 1 TO ReverbSec&
              PUT #5, n&, reverb%
            NEXT n&
Revrb:      LET sinc& = fdelay& - sdelay&
            LET ainc% = famp% - samp%
            IF sdelay& > ReverbSec& OR fdelay& > ReverbSec& THEN GOTO
RevbError:
            IF SxOnly% THEN GOTO MainLoop:
            FOR n& = 1 TO SecLength&
              LET m& = (n& - sdelay& - sinc& / SecLength& * n&)
              IF m& < 0 THEN LET m& = ReverbSec& + m&
              GET #5, m&, reverb%
              GET #4, n&, buffer%
              PUT #5, n&, buffer%
              LET reverb% = reverb% * (samp% + ainc% / SecLength& * n&) /
99
              LET buffer% = buffer% / scale% * (scale% - 1) + reverb% /
scale%
              PUT #4, n&, buffer%
            NEXT n&
            GOTO MainLoop:

InsEnd:     IF CurrentPointer& >= EndPointer& OR SxOnly% THEN GOTO EndPrg:
            FOR n& = CurrentPointer& + 1 TO EndPointer&
              PRINT #3, CHR$(0); CHR$(0);
            NEXT n&
EndPrg:     PRINT #2, "Ended on: "; DATE$; " at "; TIME$
            LOCATE 11, 1
            PRINT "Ended on:   "; DATE$; " at "; TIME$
            CLOSE
            PRINT
            PRINT "<RETURN>"
WaitKey:    IF INKEY$ <> CHR$(13) THEN GOTO WaitKey:
            SYSTEM

            END
```

ANEXO [D]

```
REMARK Estudo sobre [3^(1/27)]^9 para computador
REMARK   (C) 1992 - 1997, Carlos Alberto Gomes

REMARK Definições
OUTPUT estudo 26080 80
WAVE 0 SINE         *Onda sinusóidal
WAVE 1 SWALOW       *Onda triangular
WAVE 2 SQUARE 50    *Onda quadrada 50%
TUNING 440 1.041528 *Valor do pseudo “meio-tom”

REMARK Secção “a1”, gesto de abertura
SECTION 1250 SYNC
SOUND 1 0 0 0 0 90 9
SOUND 1 0 0 9 0 90 9
SOUND 1 0 0 -9 0 90 9
SOUND 1 0 0 18 0 90 9
SOUND 1 0 0 -18 0 90 9
SOUND 1 0 0 27 0 90 9
SOUND 1 0 0 -27 0 90 9
SOUND 1 0 0 36 0 90 9
SOUND 1 0 0 -36 0 90 9
RECORD              *Tempo parcial de 1.25''
SOUND 1 0 0 36 90 90 9
SOUND 1 0 9 36 90 90 9
SOUND 1 0 -9 36 90 90 9
SOUND 1 0 18 36 90 90 9
SOUND 1 0 -18 36 90 90 9
SOUND 1 0 27 36 90 90 9
SOUND 1 0 -27 36 90 90 9
SOUND 1 0 36 36 90 90 9
SOUND 1 0 -36 36 90 90 9
RECORD              *Tempo parcial de 2.50''
SECTION 750
MODULATION 1 0 36 -36 90 15 0 0 0 0 0 60 1
RECORD              *Tempo parcial de 3.25''
REMARK Pano de fundo para a próxima secção:
REMARK [MODULATION 1 0 -36 -36 15 0 0 0 0 60 60 1]

REMARK Secção “b1”, pano de fundo que vem de “a”
MODULATION 1 0 -36 -36 15 15 0 0 0 0 60 60 1
RECORD              *Tempo parcial de 4.00''
SOUND 0 0 18 22 0 25 1
SOUND 0 0 18 14 0 25 1
REVERB 36 36 99 99 4 CLEAR
MODULATION 1 0 -36 -36 15 15 0 0 0 0 60 60 1
RECORD              *Tempo parcial de 4.75''
SECTION 250
SOUND 0 0 22 18 25 0 1
SOUND 0 0 14 18 25 0 1
REVERB 36 36 99 99 4
MODULATION 1 0 -36 -36 15 15 0 0 0 0 60 60 1
RECORD              *Tempo parcial de 5.00''
SECTION 200
REVERB 36 36 99 99 4
MODULATION 1 0 -36 -36 15 15 0 0 0 0 60 60 1
RECORD              *Tempo parcial de 5.20''
SECTION 750
SOUND 0 0 18 23 0 25 1
```

```
SOUND 0 0 18 13 0 25 1
REVERB 36 36 99 99 4
MODULATION 1 0 -36 -36 15 15 0 0 0 0 60 60 1
RECORD              *Tempo parcial de 5.95''
SECTION 250
SOUND 0 0 23 18 25 0 1
SOUND 0 0 13 18 25 0 1
REVERB 36 36 99 99 4
MODULATION 1 0 -36 -36 15 15 0 0 0 0 60 60 1
RECORD              *Tempo parcial de 6.20''
SECTION 200
REVERB 36 36 99 99 4
MODULATION 1 0 -36 -36 15 15 0 0 0 0 60 60 1
RECORD              *Tempo parcial de 6.40''
SECTION 750
SOUND 0 0 18 21 0 25 1
SOUND 0 0 18 15 0 25 1
REVERB 36 36 99 99 4
MODULATION 1 0 -36 -36 15 15 0 0 0 0 60 60 1
RECORD              *Tempo parcial de 7.15''
SECTION 250
SOUND 0 0 21 18 25 0 1
SOUND 0 0 15 18 25 0 1
REVERB 36 36 99 99 4
MODULATION 1 0 -36 -36 15 15 0 0 0 0 60 60 1
RECORD              *Tempo parcial de 7.40''
SECTION 200
REVERB 36 36 99 99 4
MODULATION 1 0 -36 -36 15 15 0 0 0 0 60 60 1
RECORD              *Tempo parcial de 7.60''
SECTION 750
SOUND 0 0 18 24 0 25 1
SOUND 0 0 18 12 0 25 1
REVERB 36 36 99 99 4
MODULATION 1 0 -36 -36 15 15 0 0 0 0 60 60 1
RECORD              *Tempo parcial de 8.35''
SECTION 250
SOUND 0 0 24 18 25 0 1
SOUND 0 0 12 18 25 0 1
REVERB 36 36 99 99 4
MODULATION 1 0 -36 -36 15 15 0 0 0 0 60 60 1
RECORD              *Tempo parcial de 8.60''
SECTION 200
REVERB 36 36 99 99 4
MODULATION 1 0 -36 -36 15 15 0 0 0 0 60 60 1
RECORD              *Tempo parcial de 8.80''
SECTION 750
SOUND 0 0 18 20 0 25 1
SOUND 0 0 18 16 0 25 1
REVERB 36 36 99 99 4
MODULATION 1 0 -36 -36 15 15 0 0 0 0 60 60 1
RECORD              *Tempo parcial de 9.55''
SECTION 250
SOUND 0 0 20 18 25 0 1
SOUND 0 0 16 18 25 0 1
REVERB 36 36 99 99 4
MODULATION 1 0 -36 -36 15 15 0 0 0 0 60 60 1
RECORD              *Tempo parcial de 9.80''
SECTION 200
REVERB 36 36 99 99 4
MODULATION 1 0 -36 -36 15 15 0 0 0 0 60 60 1
```

```
RECORD              *Tempo parcial de 10.00''
SECTION 750
SOUND 0 0 18 25 0 25 1
SOUND 0 0 18 11 0 25 1
REVERB 36 36 99 99 4
MODULATION 1 0 -36 -36 15 15 0 0 0 0 60 60 1
RECORD              *Tempo parcial de 10.75''
SECTION 250
SOUND 0 0 25 18 25 0 1
SOUND 0 0 11 18 25 0 1
REVERB 36 36 99 99 4
MODULATION 1 0 -36 -36 15 15 0 0 0 0 60 60 1
RECORD              *Tempo parcial de 11.00''
SECTION 200
REVERB 36 36 99 99 4
MODULATION 1 0 -36 -36 15 15 0 0 0 0 60 60 1
RECORD              *Tempo parcial de 11.20''
SECTION 750
SOUND 0 0 18 19 0 25 1
SOUND 0 0 18 17 0 25 1
REVERB 36 36 99 99 4
MODULATION 1 0 -36 -36 15 15 0 0 0 0 60 60 1
RECORD              *Tempo parcial de 11.95''
SECTION 250
SOUND 0 0 19 18 25 0 1
SOUND 0 0 17 18 25 0 1
REVERB 36 36 99 99 4
MODULATION 1 0 -36 -36 15 15 0 0 0 0 60 60 1
RECORD              *Tempo parcial de 12.20''
SECTION 200
REVERB 36 36 99 99 4
MODULATION 1 0 -36 -36 15 15 0 0 0 0 60 60 1
RECORD              *Tempo parcial de 12.40''
SECTION 750
SOUND 0 0 18 26 0 25 1
SOUND 0 0 18 10 0 25 1
REVERB 36 36 99 99 4
MODULATION 1 0 -36 -36 15 15 0 0 0 0 60 60 1
RECORD              *Tempo parcial de 13.15''
SECTION 250
SOUND 0 0 26 18 25 0 1
SOUND 0 0 10 18 25 0 1
REVERB 36 36 99 99 4
MODULATION 1 0 -36 -36 15 15 0 0 0 0 60 60 1
RECORD              *Tempo parcial de 13.40''
SECTION 200
REVERB 36 36 99 99 4
MODULATION 1 0 -36 -36 15 15 0 0 0 0 60 60 1
RECORD              *Tempo parcial de 13.60''
SECTION 750
SOUND 0 0 18 18 0 50 1
REVERB 36 36 99 99 4
MODULATION 1 0 -36 -36 15 15 0 0 0 0 60 60 1
RECORD              *Tempo parcial de 14.35''
SECTION 650
SOUND 0 0 18 18 50 0 1
REVERB 36 36 99 0 4
MODULATION 1 0 -36 -36 15 15 0 0 0 0 60 60 1
RECORD              *Tempo parcial de 15.00''
SECTION 1000
MODULATION 1 0 -36 -36 15 15 0 0 0 0 60 40 1
```

```
RECORD              *Tempo parcial de 16.00''
MODULATION 1 0 -36 -36 15 15 0 0 0 0 40 20 1
RECORD              *Tempo parcial de 17.00''

REMARK Secção "c"
SECTION 500
MODULATION 1 0 -36 -18 15 10 0 0 0 0 20 10 1
RECORD              *Tempo parcial de 17.50''
MODULATION 1 0 -18 0 10 7 0 0 0 0 10 0 1
RECORD              *Tempo parcial de 18.00''
SECTION 250
SOUND 0 0 0 0 7 7 1
SOUND 0 0 0 1 0 7 1
SOUND 0 0 0 -1 0 7 1
RECORD              *Tempo parcial de 18.25''
SOUND 0 0 0 0 7 7 1
SOUND 0 0 1 1 7 7 1
SOUND 0 0 -1 -1 7 7 1
SOUND 0 0 1 2 0 7 1
SOUND 0 0 -1 -2 0 7 1
RECORD              *Tempo parcial de 18.50''
SOUND 0 0 0 0 7 7 1
SOUND 0 0 1 1 7 7 1
SOUND 0 0 -1 -1 7 7 1
SOUND 0 0 2 2 7 7 1
SOUND 0 0 -2 -2 7 7 1
SOUND 0 0 2 3 0 7 1
SOUND 0 0 -2 -3 0 7 1
RECORD              *Tempo parcial de 18.75''
SOUND 0 0 0 0 7 7 1
SOUND 0 0 1 1 7 7 1
SOUND 0 0 -1 -1 7 7 1
SOUND 0 0 2 2 7 7 1
SOUND 0 0 -2 -2 7 7 1
SOUND 0 0 3 3 7 7 1
SOUND 0 0 -3 -3 7 7 1
SOUND 0 0 3 4 0 7 1
SOUND 0 0 -3 -4 0 7 1
RECORD              *Tempo parcial de 19.00''
SOUND 0 0 0 0 7 7 1
SOUND 0 0 1 1 7 7 1
SOUND 0 0 -1 -1 7 7 1
SOUND 0 0 2 2 7 7 1
SOUND 0 0 -2 -2 7 7 1
SOUND 0 0 3 3 7 7 1
SOUND 0 0 -3 -3 7 7 1
SOUND 0 0 4 4 7 7 1
SOUND 0 0 -4 -4 7 7 1
SOUND 0 0 4 5 0 7 1
SOUND 0 0 -4 -5 0 7 1
RECORD              *Tempo parcial de 19.25''
SECTION 500
SOUND 0 0 0 0 7 7 1
SOUND 0 0 1 1 7 7 1
SOUND 0 0 -1 -1 7 7 1
SOUND 0 0 2 2 7 7 1
SOUND 0 0 -2 -2 7 7 1
SOUND 0 0 3 3 7 7 1
SOUND 0 0 -3 -3 7 7 1
SOUND 0 0 4 4 7 7 1
SOUND 0 0 -4 -4 7 7 1
```

```
SOUND 0 0 5 5 7 7 1
SOUND 0 0 -5 -5 7 7 1
SOUND 0 0 5 6 0 7 1
SOUND 0 0 -5 -6 0 7 1
RECORD              *Tempo parcial de 19.75''
SECTION 1250
SOUND 0 0 0 0 7 7 1
SOUND 0 0 1 1 7 7 1
SOUND 0 0 -1 -1 7 7 1
SOUND 0 0 2 2 7 7 1
SOUND 0 0 -2 -2 7 7 1
SOUND 0 0 3 3 7 7 1
SOUND 0 0 -3 -3 7 7 1
SOUND 0 0 4 4 7 7 1
SOUND 0 0 -4 -4 7 7 1
SOUND 0 0 5 5 7 7 1
SOUND 0 0 -5 -5 7 7 1
SOUND 0 0 6 6 7 7 1
SOUND 0 0 -6 -6 7 7 1
RECORD              *Tempo parcial de 21.00''
SOUND 0 0 0 0 7 0 1
SOUND 0 0 1 1 7 0 1
SOUND 0 0 -1 -1 7 0 1
SOUND 0 0 2 2 7 0 1
SOUND 0 0 -2 -2 7 0 1
SOUND 0 0 3 3 7 0 1
SOUND 0 0 -3 -3 7 0 1
SOUND 0 0 4 4 7 0 1
SOUND 0 0 -4 -4 7 0 1
SOUND 0 0 5 5 7 0 1
SOUND 0 0 -5 -5 7 0 1
SOUND 0 0 6 6 7 0 1
SOUND 0 0 -6 -6 7 0 1
RECORD              *Tempo parcial de 22.25''

REMARK Secção “d”
REMARK "Explosão" na gama de +27!!!
SECTION 200
MODULATION 0 0 27 27 0 70 0 0 36 36 99 60 1
RECORD              *Tempo parcial de 22.45''
SECTION 550
MODULATION 0 0 27 27 70 40 0 0 36 36 60 55 1
RECORD              *Tempo parcial de 23.00''
SECTION 1000
MODULATION 0 0 27 27 40 30 0 0 36 36 55 53 1
SOUND 1 0 -27 -18 0 20 1
RECORD              *Tempo parcial de 24.00''
MODULATION 0 0 27 27 30 30 0 0 36 36 53 52 1
SOUND 1 0 -18 -9 20 0 1
SOUND 1 0 -27 -18 0 20 1
RECORD              *Tempo parcial de 25.00''
MODULATION 0 0 27 27 30 30 0 0 36 36 52 51 1
SOUND 1 0 -18 -9 20 0 1
SOUND 1 0 -27 -18 0 20 1
RECORD              *Tempo parcial de 26.00''
MODULATION 0 0 27 27 30 30 0 0 36 36 51 50 1
SOUND 1 0 -18 -9 20 0 1
SOUND 1 0 -27 -18 0 20 1
RECORD              *Tempo parcial de 27.00''
MODULATION 0 0 27 27 30 20 0 0 36 36 50 50 1
SOUND 1 0 -18 -9 20 0 1
```

```
SOUND 1 0 -27 -18 0 20 1
RECORD              *Tempo parcial de 28.00''
MODULATION 0 0 27 27 20 10 0 0 36 36 50 50 1
SOUND 1 0 -18 -9 20 0 1
SOUND 1 0 -27 -18 0 20 1
RECORD              *Tempo parcial de 29.00''
MODULATION 0 0 27 27 10 0 0 0 36 36 50 50 1
SOUND 1 0 -18 -9 20 0 1
SOUND 1 0 -27 -18 0 20 1
RECORD              *Tempo parcial de 30.00''

REMARK Secção “e”, sucessão de glissandos
REMARK Expansão do gesto começado em “d”
SOUND 1 0 -18 -9 20 30 1
RECORD              *Tempo parcial de 31.00''
SOUND 1 0 -9 0 30 30 1
RECORD              *Tempo parcial de 32.00''
SOUND 1 0 0 9 30 30 1
RECORD              *Tempo parcial de 33.00''
SOUND 1 0 9 18 30 30 1
RECORD              *Tempo parcial de 34.00''
SOUND 1 0 18 27 30 30 1
RECORD              *Tempo parcial de 35.00''
SOUND 1 0 27 36 30 0 1
SOUND 1 0 -36 -27 0 30 1
RECORD              *Tempo parcial de 36.00''
SOUND 1 0 -27 -18 30 30 1
MODULATION 1 0 0 0 0 10 0 0 18 18 0 30 1
RECORD              *Tempo parcial de 37.00''
SOUND 1 0 -18 -9 30 30 1
MODULATION 1 0 0 0 10 0 0 0 18 18 30 30 1
RECORD              *Tempo parcial de 38.00''
SOUND 1 0 -9 0 30 30 1
RECORD              *Tempo parcial de 39.00''
SOUND 1 0 0 9 30 30 1
RECORD              *Tempo parcial de 40.00''
SOUND 1 0 9 18 30 30 1
MODULATION 1 0 0 0 0 10 0 0 -18 -18 30 30 1
RECORD              *Tempo parcial de 41.00''
SOUND 1 0 18 27 30 30 1
MODULATION 1 0 0 0 10  0 0 0 -18 -18 30 0 1
RECORD              *Tempo parcial de 42.00''
SOUND 1 0 27 36 30 0 1
SOUND 2 0 36 27 0 30 1
SOUND 2 0 -36 -27 0 30 1
RECORD              *Tempo parcial de 43.00''
SOUND 2 0 27 18 30 30 1
SOUND 2 0 -27 -18 30 30 1
RECORD              *Tempo parcial de 44.00''
SOUND 2 0 18 9 30 30 1
SOUND 2 0 -18 -9 30 30 1
RECORD              *Tempo parcial de 45.00''
SOUND 2 0 9 0 30 30 1
SOUND 2 0 -9 0 30 30 1
RECORD              *Tempo parcial de 46.00''

REMARK Secção “f”
SECTION 1000
SOUND 2 0 0 0 30 30 1
RECORD              *Tempo parcial de 47.00''
MODULATION 2 0 0 0 30 30 0 0 0 0 0 90 1
```

```
RECORD              *Tempo parcial de 48.00''
MODULATION 2 0 0 0 30 30 0 0 0 9 90 90 2
MODULATION 2 0 0 0 30 30 0 0 0 -9 90 90 2
RECORD              *Tempo parcial de 49.00''
MODULATION 2 0 0 0 30 30 0 0 9 18 90 90 2
MODULATION 2 0 0 0 30 30 0 0 -9 -18 90 90 2
RECORD              *Tempo parcial de 50.00''
MODULATION 2 0 0 0 30 30 0 0 18 27 90 90 2
MODULATION 2 0 0 0 30 30 0 0 -18 -27 90 90 2
RECORD              *Tempo parcial de 51.00''
MODULATION 2 0 0 0 30 30 0 0 27 36 90 90 2
MODULATION 2 0 0 0 30 30 0 0 -27 -36 90 90 2
RECORD              *Tempo parcial de 52.00''
MODULATION 2 0 0 0 30 30 0 0 36 36 90 0 2
MODULATION 2 0 0 0 30 30 0 0 -36 -36 90 0 2
RECORD              *Tempo parcial de 53.00''
SOUND 2 0 0 0 30 30 1
RECORD              *Tempo parcial de 54.00''
SOUND 2 0 0 9 30 30 2
SOUND 2 0 0 -9 30 30 2
RECORD              *Tempo parcial de 55.00''
SOUND 2 0 9 18 30 30 2
SOUND 2 0 -9 -18 30 30 2
RECORD              *Tempo parcial de 56.00''
SOUND 2 0 18 27 30 30 2
SOUND 2 0 -18 -27 30 30 2
MODULATION 1 0 0 0 0 10 0 0 0 0 0 50 1
RECORD              *Tempo parcial de 57.00''
SOUND 2 0 27 36 30 0 2
SOUND 2 0 -27 -36 30 0 2
MODULATION 1 0 0 0 10 10 0 0 0 0 50 80 1
RECORD              *Tempo parcial de 58.00''
MODULATION 1 0 0 0 10 10 0 0 0 0 80 0 1
RECORD              *Tempo parcial de 59.00''
SOUND 1 0 0 0 10 10 1
RECORD              *Tempo parcial de 60.00''

REMARK Secção “a2”, retorno ao gesto de abertura
SECTION 1250 SYNC
SOUND 1 0 0 0 90 90 9
SOUND 1 0 0 9 0 90 9
SOUND 1 0 0 -9 0 90 9
SOUND 1 0 0 18 0 90 9
SOUND 1 0 0 -18 0 90 9
SOUND 1 0 0 27 0 90 9
SOUND 1 0 0 -27 0 90 9
SOUND 1 0 0 36 0 90 9
SOUND 1 0 0 -36 0 90 9
RECORD              *Tempo parcial de 61.25''
SOUND 1 0 0 36 90 90 9
SOUND 1 0 9 36 90 90 9
SOUND 1 0 -9 36 90 90 9
SOUND 1 0 18 36 90 90 9
SOUND 1 0 -18 36 90 90 9
SOUND 1 0 27 36 90 90 9
SOUND 1 0 -27 36 90 90 9
SOUND 1 0 36 36 90 90 9
SOUND 1 0 -36 36 90 90 9
RECORD              *Tempo parcial de 62.50''
SECTION 750
MODULATION 1 0 36 -36 90 15 0 0 0 0 0 60 1
```

```
RECORD              *Tempo parcial de 63.25''

REMARK Secção “b2”
MODULATION 1 0 -36 -36 15 15 0 0 0 0 60 60 1
RECORD              *Tempo parcial de 64.00''
SOUND 0 0 18 22 0 25 1
SOUND 0 0 18 14 0 25 1
REVERB 72 72 99 99 3 CLEAR
MODULATION 1 0 -36 -36 15 15 0 0 0 0 60 60 1
RECORD              *Tempo parcial de 64.75''
SECTION 250
SOUND 0 0 22 18 25 0 1
SOUND 0 0 14 18 25 0 1
REVERB 72 72 99 99 3
MODULATION 1 0 -36 -36 15 15 0 0 0 0 60 60 1
RECORD              *Tempo parcial de 65.00''
SECTION 200
REVERB 72 72 99 99 3
MODULATION 1 0 -36 -36 15 15 0 0 0 0 60 60 1
RECORD              *Tempo parcial de 65.20''
SECTION 750
SOUND 0 0 18 23 0 25 1
SOUND 0 0 18 13 0 25 1
REVERB 72 72 99 99 3
MODULATION 1 0 -36 -36 15 15 0 0 0 0 60 60 1
RECORD              *Tempo parcial de 65.95''
SECTION 250
SOUND 0 0 23 18 25 0 1
SOUND 0 0 13 18 25 0 1
REVERB 72 72 99 99 3
MODULATION 1 0 -36 -36 15 15 0 0 0 0 60 60 1
RECORD              *Tempo parcial de 66.20''
SECTION 200
REVERB 72 72 99 99 3
MODULATION 1 0 -36 -36 15 15 0 0 0 0 60 60 1
RECORD              *Tempo parcial de 66.40''
SECTION 750
SOUND 0 0 18 21 0 25 1
SOUND 0 0 18 15 0 25 1
REVERB 72 72 99 99 3
MODULATION 1 0 -36 -36 15 15 0 0 0 0 60 60 1
RECORD              *Tempo parcial de 67.15''
SECTION 250
SOUND 0 0 21 18 25 0 1
SOUND 0 0 15 18 25 0 1
REVERB 72 72 99 99 3
MODULATION 1 0 -36 -36 15 15 0 0 0 0 60 60 1
RECORD              *Tempo parcial de 67.40''
SECTION 200
REVERB 72 72 99 99 3
MODULATION 1 0 -36 -36 15 15 0 0 0 0 60 60 1
RECORD              *Tempo parcial de 67.60''
SECTION 750
SOUND 0 0 18 24 0 25 1
SOUND 0 0 18 12 0 25 1
REVERB 72 72 99 99 3
MODULATION 1 0 -36 -36 15 15 0 0 0 0 60 60 1
RECORD              *Tempo parcial de 68.35''
SECTION 250
SOUND 0 0 24 18 25 0 1
SOUND 0 0 12 18 25 0 1
```

```
REVERB 72 72 99 99 3
MODULATION 1 0 -36 -36 15 15 0 0 0 0 60 60 1
RECORD              *Tempo parcial de 68.60''
SECTION 200
REVERB 72 72 99 99 3
MODULATION 1 0 -36 -36 15 15 0 0 0 0 60 60 1
RECORD              *Tempo parcial de 68.80''
SECTION 750
SOUND 0 0 18 20 0 25 1
SOUND 0 0 18 16 0 25 1
REVERB 72 72 99 99 3
MODULATION 1 0 -36 -36 15 15 0 0 0 0 60 60 1
RECORD              *Tempo parcial de 69.55''
SECTION 250
SOUND 0 0 20 18 25 0 1
SOUND 0 0 16 18 25 0 1
REVERB 72 72 99 99 3
MODULATION 1 0 -36 -36 15 15 0 0 0 0 60 60 1
RECORD              *Tempo parcial de 69.80''
SECTION 200
REVERB 72 72 99 99 3
MODULATION 1 0 -36 -36 15 15 0 0 0 0 60 60 1
RECORD              *Tempo parcial de 70.00''
SECTION 750
SOUND 0 0 18 25 0 25 1
SOUND 0 0 18 11 0 25 1
REVERB 72 72 99 99 3
MODULATION 1 0 -36 -36 15 15 0 0 0 0 60 60 1
RECORD              *Tempo parcial de 70.75''
SECTION 250
SOUND 0 0 25 18 25 0 1
SOUND 0 0 11 18 25 0 1
REVERB 72 72 99 99 3
MODULATION 1 0 -36 -36 15 15 0 0 0 0 60 60 1
RECORD              *Tempo parcial de 71.00''
SECTION 200
REVERB 72 72 99 99 3
MODULATION 1 0 -36 -36 15 15 0 0 0 0 60 60 1
RECORD              *Tempo parcial de 71.20''
SECTION 750
SOUND 0 0 18 19 0 25 1
SOUND 0 0 18 17 0 25 1
REVERB 72 72 99 99 3
MODULATION 1 0 -36 -36 15 15 0 0 0 0 60 60 1
RECORD              *Tempo parcial de 71.95''
SECTION 250
SOUND 0 0 19 18 25 0 1
SOUND 0 0 17 18 25 0 1
REVERB 72 72 99 99 3
MODULATION 1 0 -36 -36 15 15 0 0 0 0 60 60 1
RECORD              *Tempo parcial de 72.20''
SECTION 200
REVERB 72 72 99 99 3
MODULATION 1 0 -36 -36 15 15 0 0 0 0 60 60 1
RECORD              *Tempo parcial de 72.40''
SECTION 750
SOUND 0 0 18 26 0 25 1
SOUND 0 0 18 10 0 25 1
REVERB 72 72 99 99 3
MODULATION 1 0 -36 -36 15 15 0 0 0 0 60 60 1
RECORD              *Tempo parcial de 73.15''
```

```
SECTION 250
SOUND 0 0 26 18 25 0 1
SOUND 0 0 10 18 25 0 1
REVERB 72 72 99 99 3
MODULATION 1 0 -36 -36 15 15 0 0 0 0 60 60 1
RECORD              *Tempo parcial de 73.40''
SECTION 200
REVERB 72 72 99 99 3
MODULATION 1 0 -36 -36 15 15 0 0 0 0 60 60 1
RECORD              *Tempo parcial de 73.60''
SECTION 750
SOUND 0 0 18 18 0 50 1
REVERB 72 72 99 99 3
MODULATION 1 0 -36 -36 15 15 0 0 0 0 60 60 1
RECORD              *Tempo parcial de 74.35''
SECTION 650
SOUND 0 0 18 18 50 0 1
REVERB 72 72 99 0 3
MODULATION 1 0 -36 -36 15 15 0 0 0 0 60 60 1
RECORD              *Tempo parcial de 75.00''
SECTION 1000
MODULATION 1 0 -36 -36 15 15 0 0 0 0 60 40 1
RECORD              *Tempo parcial de 76.00''

REMARK Secção “g”, coda final
MODULATION 1 0 -36 -36 15 15 0 0 0 0 40 20 1
RECORD              *Tempo parcial de 77.00''
MODULATION 1 0 -36 -36 15 15 0 0 0 0 20 10 1
RECORD              *Tempo parcial de 78.00''
MODULATION 1 0 -36 -36 15 15 0 0 0 0 10 0 1
RECORD              *Tempo parcial de 79.00''
SOUND 1 0 -36 36 15 0 1
RECORD              *Tempo parcial de 80.00''

END
```

Carregue aqui para ouvir →

BIBLIOGRAFIA

CARNEIRO, ISABEL; SERRA, HELENA; FERREIRA, ODETE. *Música no futuro 5º ano*. 5ª Edição. Editorial O Livro.
CDROM, *Encarta 97*. Microsoft Corporation.
CHION, MICHAEL (1994). *Músicas, media e tecnologias*. Instituto Piaget.
COMMODORE-AMIGA, INC. *Amiga ROM kernel references manual*. Addison-Wesley Publishing Company, Inc.
DORFMAN, LEN & YOUNG, DENNIS. *Atari ST – Introduction to MIDI Programming*. Abacus Software.
GRIFFITHS, PAUL (1987). *A Música Moderna*. Jorge Jahar Editor.
INTERNET, http://www.uc.pt/
JERSE, THOMAS A. & DODGE, CHARLES. *Computer Music*. Schirmer Books.
MINISTÉRIO DA EDUCAÇÃO. *Objectivos gerais de ciclo: ensino básico 1º, 2º e 3º ciclos*. Direcção Geral do Ensino Básico e Secundário.
NICHOLS, ROGER (1986). *Messiaen*. 2nd Edition. Oxford University Press.
PONTE, JOÃO (1992). *O computador: um instrumento de educação*. 6.ª Edição. Texto Editora.
SADIE, STANLEY (1980). *The New Grove Dictionary of Music and Musicians*. Macmillan Publishers Limited, London.
VERCOE, BARRY (1986). *CSOUND - A manual for the Audio Processing System and Supporting Programs*. Media Lab, M.I.T.

---

[1] As *Ondas Martenot* foram muito utilizadas pelo compositor francês *Olivier Messiaen* (1908-92) nomeadamente na sua sinfonia *Turangalila* (1948).
[2] *Glissando* consiste na passagem entre dois sons de altura diferente passando por todos os sons de altura compreendidos entre estes.
[3] O *Órgão* de tubos é o primeiro instrumento na História da Música capaz de produzir sons multitímbricos baseado na sequência natural de harmónicos de um som através de processos de síntese aditiva (Teorema de Fourrier).
[4] Inicialmente desenvolvido para o Macintosh da Apple, o *Finale*, da *Coda Software*, é um programa de edição musical que se tornou muito popular entre os músicos, sendo usado profissionalmente na edição de partituras.
[5] Pode-se afirmar que o acto da criação musical em certas circunstâncias pode ser transformado em processos algorítmicos, nomeadamente se este se reportar a repetição de estilos preexistentes.
[6] É um acrónimo para *Musical Instrument Digital Interface*, consistindo num protocolo de comunicação de informações usado no controle de sintetizadores, tendo-se tornando um standard de facto da industria. Existem diversos programas de sequênciação MIDI, dos quais poderemos referir o *CuBase Score* ou o *Cakewalk Professional*, este último da *Twelve Tone Systems*.
[7] Chama-se BYTE ao conjunto de 8 bits.
[8] Uma vez que cada bit pode ser 0 ou 1, com oito bits é possível representar 2^8 números, ou seja, 256 quantidades numéricas diferentes.
[9] O bit mais significativo é sempre 1, ou seja, são mensagens sob a forma 1xxxxxxx.
[10] O bit mais significativo é sempre 0, ou seja, são mensagens sob a forma 0xxxxxxx, podendo representar números entre 0 e 127.
[11] O teorema de *Niquist* resume este fenómeno, afirmando que a frequência de amostragem tem de no mínimo ser duas vezes superior à frequência do som gravado.
[12] O filtro é um dispositivo que deixa passar só algumas das frequências do sinal neste injectado, podendo possuir características de *passa baixo* (corta todas as frequências acima do limite), *passa alto* (corta todas as frequências abaixo do limite), *passa banda* (corta todas as frequências abaixo do limite inferior e acima do limite superior), e ainda *rejeita banda* (corta todas as frequências acima do limite inferior e abaixo do limite superior).
[13] O teorema de Fourrier diz-nos que todo e qualquer som complexo mais não é do que o somatório de sons simples, sinusóides, cada uma com frequência e intensidade próprias.
[14] O ruído branco contem uma distribuição uniforme de todas as frequências, em quanto que no ruído rosa existe uma preponderância de uma certa gama de frequências sobre as restantes.
[15] Na prática, um filtro *passa banda* ou *rejeita banda* é construído pelo uso conjunto de um filtro *passa baixo* e de um filtro *passa alto*.
[16] Programa de escrita e sequênciação musical que permite executar pequenos trechos musicais.
[17] Não tenho conhecimento que tal *software* exista. No entanto, e apesar dos problemas levantados a nível de sistemas computacionais para reconhecimento de voz e escrita, é possível pensar, dado o desenvolvimento tido no domínio da Inteligência Artificial, que num futuro não muito longe existam recursos tecnológicos que tal o permitam.